SAND2015-6691 R

MELCOR Computer Code Manuals

Vol. 1: Primer and Users' Guide
Version 2.1.6840 2015

Date Published: August 2015

Prepared by
L.L. Humphries, R.K. Cole, D.L. Louie, V.G. Figueroa, M. F. Young

Sandia National Laboratories
Albuquerque, NM 87185-0748

Operated for the U.S. Department of Energy

H. Esmaili, Nuclear Regulatory Commission Project Manager

Prepared for Division of Systems Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC Job Code V6343

Abstract

MELCOR is a fully integrated, engineering-level computer code that models the progression of severe accidents in light water reactor nuclear power plants. MELCOR is being developed at Sandia National Laboratories for the U.S. Nuclear Regulatory Commission as a second-generation plant risk assessment tool and the successor to the Source Term Code Package. A broad spectrum of severe accident phenomena in both boiling and pressurized water reactors is treated in MELCOR in a unified framework. These include thermal-hydraulic response in the reactor coolant system, reactor cavity, containment, and confinement buildings; core heatup, degradation, and relocation; core-concrete attack; hydrogen production, transport, and combustion; fission product release and transport behavior. Current uses of MELCOR include estimation of severe accident source terms and their sensitivities and uncertainties in a variety of applications.

This publication of the MELCOR computer code manuals corresponds to MELCOR 2.1. Volume 1 contains a primer that describes MELCOR's phenomenological scope, organization (by package), and documentation. The remainder of Volume 1 contains the MELCOR User's Guides, which provide the input instructions and guidelines for each package. Volume 2 contains the MELCOR Reference Manuals, which describe the phenomenological models that have been implemented in each package. Volume 3 of this publication presents a portfolio of test and sample problems consisting of both analyses of experiments and of full plant problems.

Contents

Volume 1: Primer and Users' Guide

The following chapters are found in Volume 1 of this code manual set.

MELCOR Primer

Executive (EXEC) Package Users' Guide

Accumulator (ACC) Package Users' Guide

Burn (BUR) Package Users' Guide

Cavity (CAV) Package Users' Guide

Condenser (CND) Package Users' Guide

Control Function (CF) Package Users' Guide

Core (COR) Package Users' Guide

Control Volume Hydrodynamics (CVH) Package Users' Guide

Decay Heat (DCH) Package Users' Guide

External Data File (EDF) Package Users' Guide

Fan Cooler (FCL) Package Users' Guide

Fuel Dispersal (FDI) Package Users' Guide

Flow Path (FL) Package Users' Guide

Heat Structures (HS) Package Users' Guide

Material Properties (MP) Package Users' Guide

NonCondensible Gas (NCG) Package Users' Guide

Passive Autocatalytic Hydrogen Recombiner (PAR) Package Users' Guide

RadioNuclide (RN) Package Users' Guide

Containment Sprays (SPR) Package Users' Guide

Tabular Function (TF) Package Users' Guide

Transfer Process (TP) Package Users' Guide

Executive Summary

MELCOR is a fully integrated, engineering-level computer code whose primary purpose is to model the progression of accidents in light water reactor nuclear power plants. A broad spectrum of severe accident phenomena in both boiling and pressurized water reactors is treated in MELCOR in a unified framework. Current uses of MELCOR include estimation of fission product source terms and their sensitivities and uncertainties in a variety of applications.

The MELCOR code is composed of an executive driver and a number of major modules, or packages, that together model the major systems of a reactor plant and their generally coupled interactions. Reactor plant systems and their response to off-normal or accident conditions include:

- thermal-hydraulic response of the primary reactor coolant system, the reactor cavity, the containment, and the confinement buildings,
- core uncovering (loss of coolant), fuel heatup, cladding oxidation, fuel degradation (loss of rod geometry), and core material melting and relocation,
- heatup of reactor vessel lower head from relocated fuel materials and the thermal and mechanical loading and failure of the vessel lower head, and transfer of core materials to the reactor vessel cavity,
- core-concrete attack and ensuing aerosol generation,
- in-vessel and ex-vessel hydrogen production, transport, and combustion,
- fission product release (aerosol and vapor), transport, and deposition,
- behavior of radioactive aerosols in the reactor containment building, including scrubbing in water pools, and aerosol mechanics in the containment atmosphere such as particle agglomeration and gravitational settling, and,
- impact of engineered safety features on thermal-hydraulic and radionuclide behavior.

The various code packages have been written using a carefully designed modular structure with well-defined interfaces between them. This allows the exchange of complete and consistent information among them so that all phenomena are explicitly coupled at every step. The structure also facilitates maintenance and upgrading of the code.

Initially, the MELCOR code was envisioned as being predominantly parametric with respect to modeling complicated physical processes (in the interest of quick code execution time and a general lack of understanding of reactor accident physics). However, over the years as phenomenological uncertainties have been reduced and user expectations and demands from MELCOR have increased, the models implemented into MELCOR have become increasingly best estimate in nature. The increased speed (and decreased cost) of modern computers (including PCs) has eased many of the perceived constraints on MELCOR code development. Today, most MELCOR models are mechanistic, with capabilities approaching those of the most

detailed codes of a few years ago. The use of models that are strictly parametric is limited, in general, to areas of high phenomenological uncertainty where there is no consensus concerning an acceptable mechanistic approach.

Current uses of MELCOR often include uncertainty analyses and sensitivity studies. To facilitate these uses, many of the mechanistic models have been coded with optional adjustable parameters. This does not affect the mechanistic nature of the modeling, but it does allow the analyst to easily address questions of how particular modeling parameters affect the course of a calculated transient. Parameters of this type, as well as other numerical parameters such as convergence criteria and iteration limits, are coded in MELCOR as sensitivity coefficients, which may be modified through optional code input.

MELCOR modeling is general and flexible, making use of a "control volume" approach in describing the plant system. No specific nodalization of a system is forced on the user, which allows a choice of the degree of detail appropriate to the task at hand. Reactor-specific geometry is imposed only in modeling the reactor core. Even here, one basic model suffices for representing either a boiling water reactor (BWR) or a pressurized water reactor (PWR) core, and a wide range of levels of modeling detail is possible. For example, MELCOR has been successfully used to model East European reactor designs such as the Russian VVER and RMBK-reactor classes.

The MELCOR 2.1 code manuals are contained in three volumes. Volume 1 contains a primer that describes MELCOR's phenomenological scope, organization (by package), and documentation. The remainder of Volume 1 contains the MELCOR User's Guides, which provide the input instructions and guidelines for each package. Volume 2 contains the MELCOR Reference Manuals, which describe the phenomenological models that have been implemented in each package. Volume 3 contains a portfolio of sample demonstration problems. These problems are a combination of experiment analyses, which illustrate code model performance against data, and full plant analyses showing MELCOR's performance on larger realistic problems.

Acknowledgments

The current authors of these revised MELCOR manuals would like to acknowledge the substantial contributions made over the past 20 years by the many previous MELCOR project members. The complete list of historical participants and contributors is far too lengthy to repeat here, and no such attempt will be made. However, we would like to recognize the participation of the following individual's efforts since the last official release of the code documentation with MELCOR 1.8.5:

Larry L. Humphries	**Code Development and Testing and Project Manager**	**SNL**
Victor Figueroa	**Editor**	**SNL**
Pamela Longmire	**Editor**	**SNL[1]**
Randall K. Cole	**Code Development and Testing**	**SNL**
Michael F. Young	**Code Development and Testing**	**SNL**
David Louie	**Code Development and Testing**	**SNL**
Joonyub Jun	**Code Development and Testing and Editor**	**SNL[2]**
Nastya Mosunova	**Code Development and Testing**	**IBRAE/RAS[3]**
John Reynolds	**Configuration Management**	**SNL**
Randall O. Gauntt	**Technical Manager**	**SNL**

[1]Former Sandia National Laboratories employee. Pamela Longmire is currently employed at the United States Nuclear Regulatory Commission
[2]Former Sandia National Laboratories employee. Raymond Jun is currently employed at the Federal Authority for Nuclear Regulation in the United Arab Emirates
[3]Nuclear Safety Institute of Russian Academy of Sciences

MELCOR Primer

The MELCOR code models a wide range of physical phenomena including thermal-hydraulics; heat transfer; aerosol physics; the heatup, degradation, and relocation of reactor cores; ex-vessel debris behavior; and fission product release and transport. It was developed to model the progression of accidents in light water nuclear power plants, but many other applications are also possible.

This primer provides a starting point in understanding MELCOR and learning how to apply it. It includes an overview of the file structure, user input conventions, and the mechanics of running the code, as well as general descriptions of the phenomena modeled and of the supporting properties and utility modules that are included in MELCOR.

The information contained here is, by itself, far from sufficient to allow a new user to successfully run MELCOR. However, it provides an essential overview and introduction to the balance of the code documentation.

Contents

1 INTRODUCTION 3

2 GENERAL PROGRAM AND FILE RELATIONS 3

3 LICENSING 6

4 MELCOR PACKAGES 7

5 GETTING STARTED 10

REFERENCES 12

1 Introduction

MELCOR is a fully integrated, relatively fast-running code that models the progression of accidents in light water reactor nuclear power plants. An entire spectrum of accident phenomena is modeled in MELCOR. Characteristics of accident progression that can be treated with MELCOR include the thermal-hydraulic response in the reactor coolant system, reactor cavity, containment, and confinement buildings; core heatup and degradation; radionuclide release and transport; hydrogen production, transport, and combustion; melt ejection phenomena; core-concrete attack; heat structure response; and the impact of engineered safety features on thermal-hydraulic and radionuclide behavior.

MELCOR has been designed to facilitate sensitivity and uncertainty analyses through the use of sensitivity coefficients. Many parameters in correlations, which are hardwired constants in most codes, are implemented as sensitivity coefficients in MELCOR. Sensitivity coefficients can be changed by the user through input as discussed in the MELCOR/MELGEN Users' Guide and in the Users' Guides for each package. For example, the coefficients in a heat transfer correlation are usually assumed to be constant. However, in MELCOR the constants are coded as sensitivity coefficients that can be changed by the user to determine the sensitivity of the results to the heat transfer correlation.

The documentation of MELCOR is divided into three areas:

(1) Users' guides and

(2) Reference manuals, generally written for each package in MELCOR.

(3) Sample calculations.

The various packages are listed later in this document. Input instructions and guidelines for each package are given in the appropriate Users' Guide. The phenomenological models that have been implemented are documented in each package's reference manual. Sample problems have been provided to give the user some guidance in developing models. The purpose of this primer is to guide the uninitiated user through the extensive MELCOR documents.

2 General Program and File Relations

MELCOR is executed in two parts. The first part is called MELGEN, in which the majority of input is specified, processed, and checked. When the input checks are satisfied, a Restart File is written for the initial conditions of the calculation. The second part of MELCOR is the MELCOR program itself, which advances the problem through time based on the input to MELGEN and any MELCOR input. Graphics post processing is provided by PTFREAD, an EXCEL® add-in utility.

The files used by MELGEN and MELCOR are:

User Input	The MELGEN User Input File contains the majority of the user input defining the problem for MELCOR. MELGEN processes and checks this input and creates a Restart File for MELCOR. MELCOR relies on the Restart File for the bulk of its input. Some timestep, problem duration, and edit information is supplied via the MELCOR User Input File. The input data for MELGEN and MELCOR can be combined into a single computer file.
Output	Both MELGEN and MELCOR generate printed output, all of which is written to their respective normal Output Files. Selected information is written also to the Diagnostic, Message, and Terminal Files, as discussed below, for the convenience of the user. The Output Files echo the User Input Files with a complete listing of all user input. The MELGEN Output File gives a full listing of all processed data, including time-independent data. The MELCOR Output File contains successive edits of time-dependent data written to it at time intervals determined by the user.
Plot	The values of all MELCOR plot variables are written to the binary Plot File at time intervals determined by the user. The Plot File can be read by PTFREAD or other graphics post-processing programs.
Restart	The MELCOR database, containing all the necessary data to restart MELCOR, is written to the Restart File at time intervals determined by the user. MELGEN generates the initial Restart File containing the initial conditions of the problem set up by user input. MELCOR extends this file as required.
Message	Special messages are written to the Message File. This file is written only by MELCOR and contains the occurrence time of significant events such as vessel bottom head failure, melt ejection, hydrogen burns, etc. As a user convenience, the Message File is copied to the end of the Output File at execution termination.
Diagnostic	The Diagnostic File contains certain diagnostic messages generated by MELGEN and MELCOR, including error messages and warnings that are useful to the user. As a user convenience, the Diagnostic File is copied to the end of the Output File when there is an abnormal calculation abort.

Extended Diagnostic	The Extended Diagnostic File contains more complete diagnostic information than the Diagnostic File but retains only the latest messages. This file is most useful to the developers to trace code problems that the user cannot control.
Terminal	The Terminal File (or Batch Job “Log” File) contains direct terminal output from MELCOR giving a brief summary of the course of the calculation. In addition to special messages, the problem time, timestep, and CPU time are written to this file as requested by the user.
Stop	The user may create this file at any time during a batch MELCOR execution. If this file is present, the MELCOR calculation is terminated and data are written to the Output, Plot, and Restart Files for the last cycle.
Mail	The user may create this file at any time during a batch MELCOR execution. If this file is present, MELCOR will create a short summary of the state of the calculation and mail it to the user. The purpose of this feature is to give the user informed control over batch jobs.

The controls for these files are found in the Executive (EXEC) Package Users’ Guide. The relationship between MELGEN, MELCOR, and PTFREAD as well as the above files is shown in Figure 2.1. MACCS [1], also shown in Figure 2.1, is a program to determine off-site consequences of fission product releases to the environment.

The Diagnostic and Message Files should be closely examined following every run. The Diagnostic File contains error messages or other information that may indicate a problem with the initial conditions specified in MELGEN or with the MELCOR calculation. The user should examine the Diagnostic File carefully after every run for indications of problems in the results. The Message File contains information concerning the timing of important events such as combustion of gases, failure of the lower head, and others. This file provides a summary of the events in the calculation without having to look through the entire output file. All the messages in these two files (Message and Diagnostic) are also included in the Output file. The Message File also contains information about the Restart File.

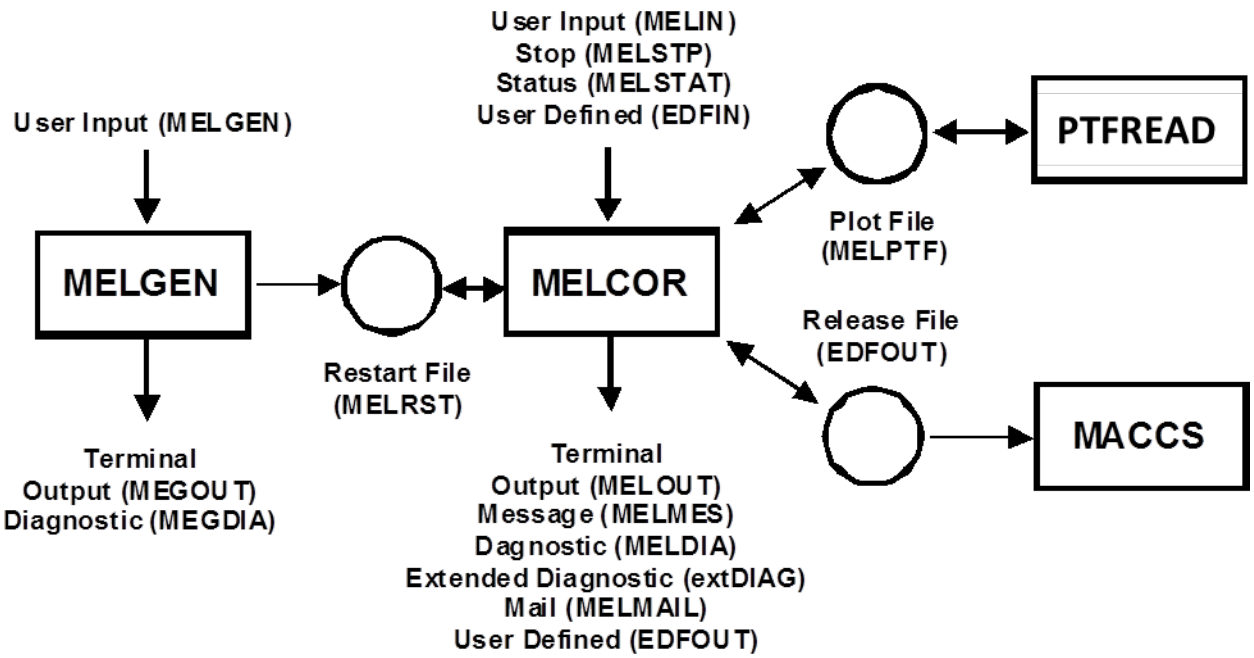


Figure 2.1 MELCOR Code and File Relations

The extended Diagnostic File contains more complete information useful to the developer when a calculation aborts. All diagnostic messages are saved in this file. Periodically, the earlier half of these messages is discarded to limit the file size.

3 Licensing

In MELCOR 2.1, licensing has been incorporated to protect the MELCOR and MELGEN executables from unauthorized changes. Each time the executables are run an attempt is made to open the *Product.key* file. If the executable does not find this file an error will be displayed and MELCOR will not run.

Follow these instructions to license MELCOR for a specific machine:

(1) Download the *license.zip* file from http://melcor.sandia.gov.

Note: Each time a license is downloaded, the MELCOR database is annotated with information that helps the NRC to track MELCOR usage statistics. The license is also encoded with information that can be used to identify its recipient. Do not transfer license files to anyone else.

(2) Extract the *license.zip* file into any directory on your hard disk.

(3) Browse to that directory and double-click on the registration script: *run_register.bat* (Windows) or *run_register.bash* (UNIX/Linux). The script calls the *register.exe* executable and pauses when it completes. The script allows you to view the registration results—if you're working at the command-line you can choose to launch the registration executable directly.

(4) The message "*Product key has been successfully bound to this computer*" will be displayed if the registration was successful. If any other message is displayed, the action is unsuccessful and MELCOR will not run. In this case, contact a member of the MELCOR team for help.

Note: License keys are time-dependent. If not registered on a computer within a short timeframe (usually a day or two), the registration process outlined here will not work. If your license key file expires before you register it, you'll need to retrieve a new license archive file and start over.

(5) Copy *calu_dll.dll* and *Product.key* to each directory containing the MELCOR and MELGEN executables. The MELCOR and MELGEN executables will not run in any folder that does not contain these files.

(6) For safety purposes, delete files *run_register.bat* (or *run_register.bash*), *register.exe* and *license.zip*. These files are not needed after the key file is registered. Running them multiple times against the same key will prevent MELCOR from running.

To license MELCOR for multiple computers, one license file is all that is needed. Simply obtain one license archive, copy it to each computer and go through steps 2 through 6 for each computer.

4 MELCOR Packages

MELCOR is composed of a number of different packages, each of which models a different portion of the accident phenomenology or program control. For example, the Control Volume Hydrodynamics (CVH) package calculates the thermal-hydraulics of control volumes, and the Core (COR) package evaluates the core behavior. Each of the packages in MELCOR is listed below with a brief description:

BUR	Burn (Combustion) of Gases. Compares conditions within control volumes against criteria for deflagrations and detonations. Initiates and propagates deflagrations involving hydrogen and carbon monoxide. Calculates burn completeness and flame speed.
CAV	Core-concrete Interactions. Models the attack on the basemat concrete by hot core materials. CORCON-MOD3 with enhanced sensitivity analysis and multi-cavity capabilities.

CF	Control Functions. Evaluates user-specified control functions and applies them to define or control various aspects of the computation such as opening and closing of valves; controlling plot, edit, and restart frequencies; defining new plot variables; etc.
CND	Condenser. Models the effects of the Isolation Condenser System and the Passive Containment Cooling System; both use heat exchangers submerged in large water pools.
COR	Core Behavior. Evaluates the behavior of the fuel and other core and lower plenum structures including heatup, candling, flow blockages, debris formation and relocation, bottom head failure, and release of core material to containment.
CVH	Control Volume Hydrodynamics. In conjunction with the FL package, evaluates mass and energy flows between control volumes.
CVT	Control Volume Thermodynamics. Evaluates the thermodynamic state within each control volume for the CVH package. No users guide is written for this package since no user input is required. However, a reference manual is written.
DCH	Decay Heat. Used by other packages to evaluate decay heat power associated with radionuclide decay.
EDF	External Data Files. Controls the reading and writing of large external data files, in close interface with CF and Transfer Process (TP) packages.
EOS	Equation of State. The CVT, H2O, and NCG packages are stored as one block of code under this name.
ESF	Engineered Safety Features. Models the thermal-hydraulics of engineered safety features that cannot be effectively modeled by building appropriate components or systems using the CVH, FL, HS, and CF packages. Currently, the ACC, CND, FCL, and PAR packages are stored under ESF; the containment sprays are modeled in the SPR package.
EXEC	Executive Package. Controls execution of MELGEN and MELCOR.
FCL	Fan Cooler. Calculates the heat and mass transfer associated with operation of the fan coolers.

FDI	Fuel Dispersal Interactions. Models ex-vessel debris relocation, heat transfer, and oxidation due to fuel-coolant interactions and high pressure melt ejection.
FL	Flow Paths. Models, in conjunction with the CVH package, the flow rates of gases and liquid water through the flow paths that connect control volumes.
H2O	Water Properties. Evaluates the water properties based on the Keenan and Keyes equation of state extended to high temperatures using the JANAF data. This set of routines is in the “EOS” code package. No user input is required.
HS	Heat Structures. Models the thermal response of heat structures and mass and heat transfer between heat structures and control volume pools and atmospheres. Treats conduction, condensation, convection, and radiation, as well as degassing of unlined concrete, and simulation of ice melt in ice condensers.
MP	Material Properties. Evaluates the physical properties of materials for other packages except for common steam and noncondensible gas properties (see H2O and NCG).
NCG	NonCondensible Gas Equation of State. Evaluates the properties of noncondensible gas mixtures using an equation of state based on the JANAF data. This set of routines is in the “EOS” code package.
PAR	Passive Autocatalytic Hydrogen Recombiner. Includes general models for modeling hydrogen recombiners in the containment rooms.
PROG	Part of MELGEN/MELCOR EXEC package separated for computer library and link purposes.
RN	Radionuclide Behavior. Models radionuclide releases, aerosol and fission product vapor behavior, transport through flow paths, and removal due to ESFs. Allows for simplified chemistry.
SPR	Sprays. Models the mass and heat transfer rates between spray droplets and control volumes.
TF	Tabular Functions. Evaluates user-selected tabular functions to define or control various aspects of the computation such as mass and energy sources; integral decay heat; plot, edit, and restart frequencies; etc.

TP	Transfer Process. Controls the transfer of core debris between various packages and the associated transfer of radionuclides within the RN package. In order to transfer core material between packages, some TP input is required, and these input are described in the COR, FDI, and CAV package Users' Guides.
UTIL	Utility Package. Contains various utilities employed by the rest of the code.

Users' guides for all packages with user input are included in Volume 1 of the MELCOR Computer Code Manuals. General input information, including the general format of input records and instructions for modification of sensitivity coefficients, appears in the EXEC Package Users' Guide. Reference manuals for the BUR, CAV, COR, CVH/FL, DCH, CVT, FCL, FDI, HS, MP, NCG/H2O, PAR, RN, and SPR packages are included in Volume 2. (Additional reference materials for several of the codes and models that have been imported into MELCOR are available separately.)

Most of these packages may be either active or inactive during a calculation. EXEC, CVH, CVT, and some of the UTIL packages are always active in any calculation. The default for most of the other packages is that they are inactive. For example, the default for the BUR package is inactive. Therefore, combustion will not be calculated to occur in MELCOR unless the package is activated. Usually, all packages are activated in the analysis of a full plant accident. The status of each package is given in the MELGEN output. Details for activating or deactivating packages are discussed in the appropriate users' guide for that package.

5 Getting Started

Experience has shown that starting with very simple thermal/hydraulic problems involving just the EXEC, CVH, and FL packages is a very good way to learn the general features of MELCOR without being overwhelmed. After the CVH and FL packages are well understood, a simple problem can be gradually increased in complexity by adding input for additional packages. A suggested order might be to next learn the CF and TF packages in conjunction with simple valve operation, followed by HS and MP packages to model simple structures such as pipe or room walls. The NCG and BUR packages could then be introduced to add more thermodynamic complexity. Actual reactor core behavior is simulated by the COR package, which could then be added along with the DCH package using the ANS decay curve option.

Addition of the ESF and SPR packages can be attempted at any time once the basic thermal-hydraulics are understood. Inclusion of the RN package should probably await a thorough understanding by the user of all the aforementioned thermal-hydraulics-oriented packages. Finally, the FDI and CAV packages can be added, along with the TP package to control the interfaces between them and with the COR package.

A set of small test problems is available to aid in user training and code testing. New users might profit from study of these input sets [2]. A growing set of assessment reports is available for more complex situations [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. A limited distribution of sample input decks can be retrieved from the download site (direct your browser to http://melcor.sandia.gov and click on the Downloads link). MELCOR Workshop content archives are also available at this site. These contain input decks illustrating content shared during workshop sessions. Additionally, Volume 3 of this document presents a portfolio of demonstration problems, including input decks and plotted code results that illustrate the use of MELCOR on a variety of real problems. The demonstration problems are a combination of relevant experiments and full plant analyses.

Once the user is comfortably familiar with the basics of MELCOR input, the demonstration problem supplied with MELCOR distribution is useful to study to see how the packages interact with each other. However, be aware that this sample problem is simply designed to exercise all MELCOR packages through a complete transient in a short calculation and that it is not intended to represent any real system nor recommended approaches to modeling real systems.

MELCOR was originally designed to be run with relatively large timesteps and coarse nodalizations for most large integral plant calculations. For a simple full reactor plant model, a base case nodalization, including reactor coolant system, containment, and auxiliary buildings, will involve typically 15 to 25 control volumes, 100 to 200 heat structures, 3 or 4 core rings, and 10 to 15 core/lower plenum levels. Sensitivity studies on a particular aspect of the sequence may dictate use of finer nodalization for some systems. Complex plant models that treat in-vessel and RCS natural circulation could make use of significantly more detailed CVH nodalization in these regions.

User-imposed maximum timesteps should range from 5 to 10 seconds during the portion of an accident sequence dominated by in-vessel thermal-hydraulics and core melt progression and from 20 to 30 seconds during the portion dominated by containment thermal-hydraulics and molten core-concrete interactions. Although many MELCOR models will reduce the timestep to lower values when needed, very rapid phenomena, certain phenomenological events, or numerical problems encountered by the code may necessitate use of a smaller maximum timestep supplied by the user for portions of the transient. As a result, the current code is somewhat dependent on the skill of the user to select proper timesteps until additional automatic timestep controls are developed.

References

1. D. I. Chanin, J. L. Sprung, L. T. Ritchie, and H-N Jow, MELCOR Accident Consequence Code System (MACCS), SAND86-1562, NUREG/CR-4691, Sandia National Laboratories, Albuquerque, NM (February 1990).

2. L. N. Kmetyk, MELCOR Assessment: Gedanken Problems, Volume 1, SAND92-0762, NPRW-SA92-22, Sandia National Laboratories, Albuquerque, NM (January 1993).

3. L. N. Kmetyk, MELCOR 1.8.1 Assessment: LACE Aerosol Experiment LA4, SAND91-1532, Sandia National Laboratories, Albuquerque, NM (September 1991).

4. L. N. Kmetyk MELCOR 1.8.1 Assessment: FLECHT SEASET Natural Circulation Experiments, SAND91-2218, Sandia National Laboratories, Albuquerque, NM (December 1991).

5. L. N. Kmetyk, MELCOR 1.8.1 Assessment: ACRR Source Term Experiments ST-1/ST-2, SAND91-2833, Sandia National Laboratories, Albuquerque, NM (April 1992).

6. L. N. Kmetyk, MELCOR 1.8.1 Assessment: LOFT Integral Experiment LP-FP-2, SAND92-1373, Sandia National Laboratories, Albuquerque, NM (December 1992).

7. L. N. Kmetyk, MELCOR 1.8.1 Assessment: Marviken-V Aerosol Transport Tests ATT-2b/ATT-4, SAND92-2243, Sandia National Laboratories, Albuquerque, NM (January 1993).

8. R. J. Gross, MELCOR 1.8.1 Assessment: PNL Ice Condenser Aerosol Experiments, SAND92-2165, Sandia National Laboratories, Albuquerque, NM (June 1993)

9. R. J. Gross, S. L. Thompson, and G. M. Martinez, MELCOR 1.8.1 Calculations of ISP31: The CORA-13 Experiment, SAND92-863, Sandia National Laboratories, Albuquerque, NM (June 1993).

10. L. N. Kmetyk, MELCOR 1.8.2 Assessment: IET Direct Containment Heating Tests, SAND93-1475, Sandia National Laboratories, Albuquerque, NM (September 1993).

11. T. J. Tautges, MELCOR 1.8.2 Assessment: The DF-4 BWR Damaged Fuel Experiment, SAND93-1377, Sandia National Laboratories, Albuquerque, NM (October 1993).

12. L. N. Kmetyk, Air Ingression Calculations for Selected Plant Transients Using MELCOR, SAND93-3808, Sandia National Laboratories, Albuquerque, NM (January 1994).

13. L. N. Kmetyk et al., MELCOR 1.8.2 Assessment: Surry PWR TMLB (with a DCH Study), SAND93-1899, Sandia National Laboratories, Albuquerque, NM (February 1994).

14. T. J. Tautges, MELCOR 1.8.2 Assessment: The MP-1 and MP-2 Late Phase Melt Progression Experiments, SAND94-0133, Sandia National Laboratories, Albuquerque, NM (May 1994).

15. L. N. Kmetyk, MELCOR 1.8.2 Assessment: GE Large Vessel Blowdown and Level Swell Experiments, SAND94-0361, Sandia National Laboratories, Albuquerque, NM (July 1994).

16. J. J. Carbajo, Comparison of MELCOR Modeling Techniques and Effects of Vessel Water Injection on a Low-Pressure, Short-Term, Station Blackout at the Grand Gulf Nuclear Station, ORNL/TM-12771, Oak Ridge National Laboratory (June 1995).

17. S. R. Greene et al., The Response of BWR Mark II Containments to Station Blackout Severe Accident Sequences, NUREG/CR-5565, Oak Ridge National Laboratory (May 1991).

18. S. R. Greene et al., The Response of BWR Mark III Containments to Short-Term Station Blackout Severe Accident Sequences, NUREG/CR-5571, Oak Ridge National Laboratory (June 1991).

19. J. J. Carbajo, Severe Accident Source Term Characteristics for Selected Peach Bottom Sequences Predicted by the MELCOR Code, NUREG/CR-5942, Oak Ridge National Laboratory (September 1993).

Accumulator (ACC) Package
Users' Guide

A simple accumulator model was added to MELCOR to simplify the input for this engineered safety feature, add flexibility in modeling, and to improve performance and numerics. Note that the accumulator can be alternately modeled using a set of control functions and control volumes, but this model was designed to simplify the modeling for the user. Furthermore, the extreme expansion of the accumulator gas can lead to temperatures below 273.15 K, which can lead to a fatal error in MELCOR causing an error in the water properties tables because ice formation is not modeled, leading to the practice of modeling the accumulator as a source of mass and enthalpy in the control volume. Note however, that this error test can be over-ruled for dry atmospheres (no water present) by use of the CVH_ALLOWCOLDATM record (see CVH UG).

The separate object "accumulator" with all necessary data for accumulator modeling has been developed and placed in ESF package. This object can be used as a source of liquid in the control volume. Input parameters for the accumulator are set on four input records. The simplified model to calculate liquid injection is implemented in the code.

More detailed information on modeling accumulators using this package can be found in the ACC Package Reference Manual.

Contents

CONTENTS **2**

1 INTRODUCTION **3**

2 INPUT REQUIREMENTS **3**

2.1 MELGEN Input 3

ACC_ID – Accumulator and Connected Control Volume Names 3
ACC_SLP – Surge Line Parameters 4
ACC_PAR – Accumulator Parameters 5
ACC_LAG – Lag Accumulator Parameters 6

3 PLOT VARIABLES **7**

4 CONTROL FUNCTION ARGUMENTS **7**

5 EXAMPLE INPUT **8**

5.1 Example MELGEN Input 8

6 ACC MODEL OUTPUT **8**

7 DIAGNOSTIC AND ERROR MESSAGES **9**

1 Introduction

A simple accumulator model was added to MELCOR to simplify the input for the engineered safety feature, add flexibility in modeling, and to improve performance and numerics. Note that the accumulator can be modeled using a set of control functions and control volumes, but this model was designed to simplify the modeling for the user. Furthermore, the extreme expansion of the accumulator gas can lead to the temperature drop below 273.15 K, which can lead to a fatal error in MELCOR model since it causes an error in the water properties tables because of ice formation is not modeled, leading to the practice of modeling the accumulator as a source of mass and enthalpy in the control volume. Note however, that this error test can be over-ruled for dry atmospheres (no water present) by use of the CVH_ALLOWCOLDATM record (see CVH UG).

The separate object "accumulator" with all necessary data for accumulator modeling has been developed and placed in ESF package. This object can be used as a source of liquid in the control volume. Input parameters for the accumulator are set on four input records. The simplified model to calculate liquid injection is implemented in the code.

2 Input Requirements

2.1 MELGEN Input

This section provides input requirements for the ACC package and includes a short description of the input quantities, their units, and default values, if any. Further description of the input variables and their meaning in the models can be found in the ACC Package Reference Manual.

Input for the ACC model must be inside the data block for the ESF package, which begins with the character string ESF_INPUT.

ACC_ID – Accumulator and Connected Control Volume Names

Required for Activation of the ACC Model

This record specifies a user-supplied name for the accumulator and connected control volume. This record is required. The following characters fields (limited to 16 characters) must be present.

(1) ACCNAM – accumulator name.
(type = character*16, default = none)

(2) CVHNAM
Name of control volume associated with accumulator.

(type = character*16, default = none)

Initialization Characteristics of the ACC Model

This input record acts as a flag to activate the ACC model. If entered, then all remaining ACC model input cards are processed. If the input record card is absent, no memory is allocated for the ACC model variables and the plot variables unique to the ACC model are not defined.

Example

```
ACC_ID  Accumulator    CV120
```

ACC_SLP – Surge Line Parameters

Required

This record specifies the parameters for the surge line connecting accumulator with the control volume (cold leg). This record is required. The first two real fields must be present, the third through sixth fields are optional (they defines the fraction of the surge line open area and the way of friction and loss coefficients definition).

(1) DIAM

The surge line diameter (value must be positive).

(type = real, default = none, units=m)

(2) LEN

The surge line length (value must be positive).

(type = real, default = none, units=m)

(3) IKEY

The user may choose one of the following options defining fraction of surge line open area:

(a)	-1 or CF,	control function
(b)	1 or TF,	tabular function
(c)	0 or CONST,	constant value

(type = integer/character*5, default = CONST)

Optionally followed by:

if IKEY = 1, -1, CF or TF

(4a) NameCFTF – tabular or control function name

(type =character*16, default = none, units = none)

or if IKEY = 0 or IKEY = CONST

(4b) FRACT – the fraction value
(type =real, default = 1., units = none)

(5) IFRIC – option for friction coefficient value:

1 or MODEL,	friction factor is calculated from the Colebrook-White equation;
0 or CONST,	constant value defined by user is used.

(type = integer / character*5, default = CONST)

(6) FRIC – constant friction coefficient value.
This field is required if and only if IFRIC = 0 or CONST. The value must be non-negative.
(type = real, default = none, units = none)

(7) IFLRES – option for form loss coefficient value.
The user may choose one of the following options:

-1 or CF,	control function
0 or CONST,	constant value

(type = integer / character*5, default = CONST)

Optionally followed by:

if IKEY = -1 or CF

(8a) NameCF –control function name
(type =character, default = none, units = none)

or if IKEY = 0 or IKEY = CONST

(8b) FLRES – the form loss coefficient value. The value must be positive.
(type =real, default = 5.65, units = none)

ACC_PAR – Accumulator Parameters

Required

This record specifies the necessary parameters for the accumulator. This record is required. The first 4 fields must be present because they are required fields, the fifth and sixth fields (defining the expansion coefficients option) are optional

(1) VN2 – the initial volume of N2 in the accumulator. The value must be positive.
(type = real, default = none, units = m3)

(2) VH2O – the initial water volume in the accumulator.
The value must be positive.
(type = real, default = none, units = m3)

(3) PAC – the initial pressure in the accumulator.
The value must be positive.
(type = real, default = none, units = Pa)

(4) TH2O – the temperature of the water in the accumulator.
The value must be greater than 0ºC.
(type = real, default = none, units = K)

(5) EXCOEF – the expansion coefficient option.
User must choose one of the following options:

1 or ADIAB,	adiabatic
2 or THERM,	isothermal
3 or USER,	user-defined

(type = integer or character, default = 1, units = none)

The following field is required if and only if EXCOEF = 3 or USER.

(6) GAMMA – the index of power for the expansion coefficient option.
(type = real, default = none, units = none)

ACC_LAG – Lag Accumulator Parameters

Required

This record specifies the necessary parameters if user wants to use lag control component for the liquid velocity in the accumulator. If this record is not set, the lag transform is not used. Both fields must be present if the record is input. The user should define the A1 and A2 constant in the following transform equation used for liquid velocity:

$$Y^{n+1} = \frac{Y^n(1-\frac{\Delta t}{2A_2}) + A_1\left(V^n + V^{n+1}\right)\frac{\Delta t}{2A_2}}{1+\frac{\Delta t}{2A_2}}$$

Eq. 1

where A1 is a scaling factor and A2 is a lag time, V is a liquid velocity calculated according to accumulator simplified model and Y is a transform liquid velocity. Note that

transferred mass and enthalpy will be defined through the transformed velocity value (Y).

(1) A1 – the scaling factor.
(type = real, default = none, units = none)

(2) A2 – the lag time (nonzero value)
(type = real, default = none, units = none)

3 Plot Variables

The plot variables currently included in the accumulator model are listed below, along with a brief description. Note that control function arguments (identical in definition to these plot variables but different in format) are described in the following section.

ESF-ACC-RAT.n	Liquid mass flow rate for accumulator index n. (units = kg/s)
ESF-ACC-MAS.n	Total liquid mass injected from accumulator index n. (units = kg)
ESF-ACC-PRS.n	Accumulator index n pressure (units = Pa)
ESF-ACC-REM.n	Total liquid mass remaining in the accumulator index n. (units = kg)
ESF-ACC-ENG.n	Integral of energy flow out for the accumulator index n. (units = J)

4 Control Function Arguments

The control function arguments currently included in the accumulator model are listed below, along with a brief description. Note that plot variables (identical in definition to these control function arguments but different in format) are described in the previous section.

ESF-ACC-RAT(ACCNAM)	Liquid mass flow rate for accumulator name ACCNAM. (units = kg/s)

ESF-ACC-MAS(ACCNAM)	Total liquid mass injected from accumulator name ACCNAM. (units = kg)
ESF-ACC-PRS(ACCNAM)	Accumulator name ACCNAM pressure. (units = Pa)
ESF-ACC-REM(ACCNAM)	Total liquid mass remaining in the accumulator name ACCNAM. (units = kg)
ESF-ACC-ENG(ACCNAM)	Integral of energy flow out for the accumulator name ACCNAM. (units = J)

5 Example Input

The following are sample MELGEN input records for the ACC Package. An example of MELCOR input deck follows.

5.1 Example MELGEN Input

```
!* Block: ESF data
ESF_INPUT
!
ACC_ID 'ACC1'  'CV240-CLA ACCUM'
ACC_SLP  0.26662 32.0 CF 'TEST1' CONST 0.05 CF 'TEST2'
ACC_PAR  14.8664  26.193  4.2403E6 322.04 ADIAB
!
ACC_ID 'ACC2' 'CV340-CLB ACCUM'
ACC_SLP  0.26662 32.0 CONST 1.0 CONST 0.05 CONST 5.65
ACC_PAR  14.8664  26.193  4.2403E6 322.04 ADIAB
!
ACC_ID 'ACC3' 'CV440-CLC ACCUM'
ACC_SLP  0.26662 32.0 CONST 1.0 CONST 0.05 CONST 5.65
ACC_PAR  14.8664  26.193  4.2403E6 322.04 ADIAB
```

6 ACC Model Output

MELGEN output

```
***** ACCUMULATORS SETUP EDIT *****

TOTAL NUMBER OF ACCUMULATORS =     3

  ACCUMULATOR NAME                    = ACC1
  CONNECTED CONTROL VOLUME NAME       = CV240-CLA ACCUM
  SURGE LINE DIAMETER                 =  0.26662E+00
  SURGE LINE LENGTH                   =  0.32000E+02
  INITIAL N2 VOLUME                   =  0.14866E+02
```

```
INITIAL WATER VOLUME                =  0.26193E+02
ACCUMULATOR INITIAL PRESSURE        =  0.42403E+07
WATER TEMPERATURE                   =  0.32204E+03
THE EXPANSION COEFFICIENT OPTION    = ADIAB
THE INDEX OF POWER                  =  0.10000E+01
SURGE LINE OPEN AREA OPTION         = CONST
SURGE LINE OPEN AREA FRACTION       =  0.10000E+01
FRICTION COEFFICIENT OPTION         = CONST
FRICTION COEFFICIENT VALUE          =  0.50000E-01
FORM LOSS COEFFICIENT OPTION        = CONST
FORM LOSS COEFFICIENT VALUE         =  0.56500E+01
```

MELCOR output file

```
*********************************
 *     ACCUMULATOR EDIT       *
 *********************************


                         ACCUMULATORS

ACCUMULATOR       CONTROL VOLUME    INJECTED MASS     INJECTED ENTHALPY
                                         KG                  J

ACC1              CV240-CLA ACCUM     0.68055E+03      0.14170E+09
ACC2              CV340-CLB ACCUM     0.41869E+03      0.87198E+08
ACC3              CV440-CLC ACCUM     0.41835E+03      0.87126E+08
```

7 Diagnostic and Error Messages

Diagnostic messages are printed to indicate the time at which each accumulator starts to inject and when the accumulator is exhausted. Other diagnostics are provided to indicate improper input or when values exceed ranges.

```
<Diagnostic Message>  Time= 7.9038E+00 Dt= 5.0000E-03 Cycle= 8401 (CVH)
  Event message for accumulator ACC1
  ****************************************
  * ACCUMULATOR INJECTION BEGINS AT TIME = 7.90382E+00 *
  ****************************************

<Diagnostic Message>  Time= 5.9594E+01 Dt= 1.0000E-02 Cycle= 14278 (CVH)
  Event message for accumulator ACC1
  ****************************************
  * ACCUMULATOR EXHAUSTED AT TIME = 5.95936E+01 *
  ****************************************
```

Burn (BUR) Package
Users' Guide

The MELCOR Burn (BUR) package models the combustion of gases in control volumes. The models consider the effects of burning on a global basis without modeling the actual reaction kinetics or tracking the actual flame front propagation. The BUR package models are based on the deflagration models and diffusion flame model in the HECTR 1.5 code.

This Users' Guide describes the input to the BUR package, including a brief description of the models employed, the input format, sample input, discussion of the output, sensitivity coefficients, plot variables, and control function arguments. Details on the models can be found in the BUR Package Reference Manual.

Contents

1 INTRODUCTION 4

2 USER INPUT 4

2.1 MELGEN Input 5

BUR_INPUT – Activation Record 5
BUR_IGN – Ignition Parameters 5
BUR_IGN01 – Ignition Parameters 6
BUR_DET – Detonation Parameters 6
BUR_COM – Combustion Completeness and Propagation Parameters 7
BUR_COM01 – Combustion Completeness and Propagation Parameters 8
BUR_PLT – Plot Edit Control 8
BUR_DIF – Diffusion Flame Activation and Burn Parameters 9
BUR_CF – Diffusion Flame Combustion Completeness Parameters. 10
BUR_BRT – Burn Rate and Propagation Speed Parameters 11
BUR_CC – Combustion Completeness Option 13
BUR_FS - Flame Speed Option 15

2.2 MELCOR Input 17

BUR_TIM – Burn Timestep Information 17

3 SENSITIVITY COEFFICIENTS 17

3.1 BUR Sensitivity Coefficients 18

C2200 - Flame Speed Parameters 18
C2201 - Maximum Overshoot of Ignition Limit 20
C2202 - Combustion Completeness Parameters 20
C2203 - Minimum Debris Temperature to Invoke DCH Parameters 21
BUR_SC – BUR Sensitivity Coefficients Input Record 21

4 PLOT VARIABLES 22

5 CONTROL FUNCTION ARGUMENTS 24

6 ILLUSTRATIVE INPUT 26

6.1 MELGEN Input 26

6.2 MELCOR Input 27

7 DISCUSSION OF OUTPUT 27

8 ERROR MESSAGES 28

APPENDIX 1 29

1 Introduction

The Burn (BUR) package models the combustion of gases in control volumes. The bulk burn models consider the effects of burning premixed gases without modeling the actual reaction kinetics or tracking the actual flame front propagation. A simple diffusion flame model (introduced in MELCOR 1.8.5) allows for the burning of hydrogen-rich mixtures upon entry to volumes containing oxygen. The models in the BUR package are based on the models in the HECTR 1.5 code. The only significant modifications made were to provide more direct user control of the models through the implementation of sensitivity coefficients and to include optional model parameters that are used to override the nominal parameters in control volumes in which direct containment heating (DCH) is occurring. Note that the diffusion flame model can be used to model the burning during DCH while leaving the bulk burn parameters at their nominal values.

Deflagrations are ignited if the mole fraction composition in a control volume satisfies a form of LeChatelier's formula. Tests for sufficient H_2 and O_2 are performed, as well as an inerting test for the presence of excessive diluents (H_2O and CO_2). Deflagrations are propagated into adjoining control volumes if additional tests for the H_2 and CO mole fractions in those volumes are satisfied and if the flow path is open to gas flow (i.e., the area is nonzero and the flow path is not covered by water). There is, however, no test for check valves.

The combustion rate is determined by the flame speed, the volume characteristic dimension, and the combustion completeness. The flame speed and combustion completeness can each be input as constant values, or they may be calculated from user-defined control functions or the default HECTR correlations. The latter are derived from experimental data and depend on combustible and diluent gas concentrations.

For user convenience, the BUR package also prints messages to warn the user when the detonability criteria are satisfied in a control volume. However, only deflagrations are modeled; detonations are merely flagged.

Note: The gases H_2, CO, CO_2, and O_2 must be defined in the NonCondensible Gas (NCG) package whenever the BUR package is active. Steam (H_2O) is automatically present for all MELCOR calculations; therefore, no special action needs to be taken to include it in a calculation.

2 User Input

The input for the BUR package consists of combustion parameters in the MELGEN and MELCOR input and timestep values for the start of each burn in the MELCOR input. If a BUR record is included in the user's input, all entries for that record must be included, except for cases noted in the following descriptions.

2.1 MELGEN Input

BUR_INPUT – Activation Record

Required

This record activates the BUR package.

(1) IACTV

Activation Parameter

Optional

(a) 0 or ACTIVE

BUR package is active.

(b) 1 or NOTACTIVE

BUR package is not active,

(type = integer, default = 0, units = none)

Example

```
BUR_INPUT NOTACTIVE
```

BUR_IGN – Ignition Parameters

Optional

The ignition parameters for deflagrations are input to this record. Input for all six entries must be specified if this record is included. See Section 2.2 of the Burn (BUR) Package Reference Manual.

(1) XH2IGN

H_2 mole fraction limit for ignition without igniters.

(type = real, default = 0.10, units = none)

(2) XCOIGN

CO mole fraction limit for ignition without igniters.

(type = real, default = 0.167, units = none)

(3) XH2IGY

H_2 mole fraction limit for ignition with igniters.

(type = real, default = 0.07, units = none)

(4) XCOIGY

CO mole fraction limit for ignition with igniters.

(type = real, default = 0.129, units = none)

(5) XO2IG

Minimum O_2 mole fraction for ignition.

(type = real, default = 0.05, units = none)

(6) XMSCIG

Maximum H_2O plus CO_2 mole fraction for ignition.

(type = real, default = 0.55. units = none)

Example

```
BUR_IGN    0.098    0.16    0.068    0.12    0.48    0.098
```

BUR_IGN01 – Ignition Parameters

Optional (can appear if and only if BUR_IGN record is input)

The ignition parameters for deflagrations are input to this record. Input for all four entries must be specified if this record is included. See Section 2.2 of the Burn (BUR) Package Reference Manual.

(1) XH2DCH

H_2 mole fraction limit for ignition during DCH.

(type = real, default = XH2IGY, units = none)

(2) XCODCH

CO mole fraction limit for ignition during DCH.

(type = real, default = XCOIGY, units = none)

(3) XO2DCH

O_2 mole fraction limit for ignition during DCH.

(type = real, default = XO2IG, units = none)

(4) XINDCH

H_2O plus CO_2 mole fraction limit for ignition during DCH.

(type = real, default = XMSCIG, units = none)

Example

```
BUR_IGN01    0.098    0.16    0.068    0.124
```

BUR_DET – Detonation Parameters

Optional

The parameters to indicate a detonation are input to this record. Input for all three entries must be specified if this record is included. See Section 2.7 of the BUR Package Reference Manual.

(1) XH2DET

Minimum H_2 mole fraction for detonation.

(type = real, default = 0.14, units = none)

(2) XO2DET

Minimum O_2 mole fraction for detonation.

(type = real, default = 0.09, units = none)

(3) XH2ODT

Maximum H_2O mole fraction for detonation.

(type = real, default = 0.30, units = none)

Example

```
BUR_DET    0.14    0.09    0.3
```

BUR_COM – Combustion Completeness and Propagation Parameters

Optional

The parameters controlling the extent of combustion and propagation to connected volumes are input to this record. Input for all eight entries must be specified if this record is included. See Sections 2.3 and 2.6 of the BUR Package Reference Manual.

(1) XH2CC

H_2 mole fraction value for calculating combustion completeness.

(type = real, default = 0.08, units = none)

(2) XCOCC

CO mole fraction value for calculating combustion completeness.

(type = real, default = 0.148, units = none)

(3) XH2PUP

H_2 mole fraction limit for upward propagation.

(type = real, default = 0.041, units = none)

(4) XCOPUP

CO mole fraction limit for upward propagation.

(type = real, default = 0.125, units = none)

(5) XH2PHO

H_2 mole fraction limit for horizontal propagation.

(type = real, default = 0.06, units = none)

(6) XCOPHO

CO mole fraction limit for horizontal propagation.

(type = real, default = 0.138, units = none)

(7) XH2PDN

H_2 mole fraction limit for downward propagation.

(type = real, default = 0.09, units = none)

(8) XCOPDN

CO mole fraction limit for downward propagation.

(type = real, default = 0.150, units = none)

Example

```
BUR_COM  0.078  0.14  0.039  0.12  0.057  0.13  0.089  0.14
```

BUR_COM01 – Combustion Completeness and Propagation Parameters

Optional (can appear if and only if record BUR_COM is input)

The parameters controlling the extent of combustion and propagation to connected volumes are input to this record. Input for all two entries must be specified if this record is included. See Sections 2.3 and 2.6 of the BUR Package Reference Manual.

(1) XH2CCD

Value of XH2CC during DCH.

(type = real, default = XH2CC, units = none)

(2) XCOCCD

Value of XCOCC during DCH.

(type = real, default = XCOCC, units = none)

Example

```
BUR_COM01    0.078    0.147
```

BUR_PLT – Plot Edit Control

Optional

The plot edits requested by the BUR package are controlled by this input.

(1) IFLAGS

Plot controlling flag.

(a) 1 or CF

Plot edit will be requested at start and end of each burn if control function CFLNAM is true; no BUR plot edits will be requested if it is false.

(b) -1 or BYREQ,

Plot edits will be requested at the start and end of each burn.

(c) -2 or NOPLOT

No plot edits will be requested by the BUR package.

(type = integer/character*6, default = -1, units = none)

Next field is required if and only if IFLAGS = 1 or IFLAGS = CF:

(1a) CFLNAM

Name of logical control function that plot edit depends on.

(type = character*16, default = none)

Example

```
BUR_PLT     CF     CF92
```

BUR_DIF – Diffusion Flame Activation and Burn Parameters

Optional

This record activates the diffusion flame model and specifies the burn parameters for the flame. The default values are set to essentially burn most of the incoming H_2 and CO in a flow path during DCH and are quite different than the typical settings for the bulk burn model. If only the first field is present, the model is activated with default values; otherwise, all seven fields must be present.

(1) IACTF

Diffusion flame model activation flag. Default is 0 (off), set to 1 (on) to activate model.

(a) 0 or OFF

Off

(b) 1 or ON

On

(type = integer / character*3, default = 0, units = none)

(BUR.IACTF)

The following six fields are required if and only if IACTF = 1 or IACTF = ON:

(2) XH2FLM

Minimum H_2 mole fraction of the flow path to burn. This should be set lower than the usual bulk burn default for realistic simulation (default is 0.0 and up to about 0.03 is reasonable).

(type = real, default = 0.07, units = none)

(3) XCOFLM

Minimum CO mole fraction of the flow path to burn. Default is typical of bulk burn during DCH.

(type = real, default = 0.129, units = none)

(4) XO2FLM

Minimum O_2 mole fraction in downstream control volume to burn.

(type = real, default = 0.01, units = none)

(5) XSCFLM

Maximum inerting gas concentration in downstream control volume (steam + CO_2) before burn stops.

(type = real, default = 0.9, units = none)

(6) XH2CCF

H_2 mole fraction to use in the LeChatelier formula for the combustion completeness of the diffusion flame. This is set to the DCH bulk burn value by default.

(type = real, default = 0.08, units = none)

(7) XCOCCF

CO mole fraction to use in the LeChatelier formula for the combustion completeness of the diffusion flame. This is set to the DCH bulk burn value by default.

(type = real, default = 0.148, units = none)

Example

```
BUR_DIF    ON    0.05    0.12    0.03    0.8    0.085    0.15
```

BUR_CF – Diffusion Flame Combustion Completeness Parameters.

Optional

This record specifies the diffusion flame combustion completeness parameters for different flow paths. A constant combustion completeness (CC) factor may be specified. Alternatively, the HECTR correlation is available, although the validity of the HECTR correlation for a diffusion flame is unknown.

(1) NUMCF

Number of user-input data strings determining the diffusion flame combustion completeness parameters for a flow path, dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMCF:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) IFLNUM

Flow path name. Putting –1 as the flow path name will cause the following values to be used for all flow paths except those redefined on subsequent BUR_CF tabular record lines.

(type = character*16 / integer, default = none, units = none)

(3) ICCFLG

Option to use for combustion completeness. For all options, the completeness must satisfy 0.0 ≤ CC ≤ 1.0.

(a) -1 or CONST

Use constant CC.

(b) 0 or HECTR

Use HECTR correlation.

(type = integer / character*5, default = 0, units = none)

(4) CC

Combustion completeness, 0.0 ≤ CC ≤ 1.0. Required if and only if ICCFLG = -1 or CONST.

(type = real, default = 0.0, units = none)

Example

```
BUR_CF  1 ! N FL NAME          OPTION COMB.COMPLETENESS
           1 FL171             CONST         0.8
```

BUR_BRT – Burn Rate and Propagation Speed Parameters

Optional

The parameters determining the burn rate and the speed of propagation are on this record series. If only the first two fields are present on the table line, then default values will be used for the remaining entries. See Sections 2.4, 2.5, and 2.6 of the BUR Package Reference Manual.

(1) NUMBRT

Number of user-input data strings determining the burn rate and the speed of propagation, dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMBRT:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) ICVNUM

User control volume name.

(type = character*16, default = none, units = none)

(3) IGNTR

Igniter variable.

(a) -1 or CF,
IGNTR variable given by REAL control function name given in additional field (3a).

(b) 0 or NOTACT
Igniter not active in control volume.

(c) 1 or ACT
Igniter is active in control volume.

(d) 86 or PROH
Burning prohibited in control volume.

(type = integer / character*6, default = 0, units = none)

(3a) CFNAME

REAL control function name. Required if and only if IGNTR = -1 or CF.. The CF value indicates when the igniter is activated in the CV where 0 = not active, 1 = active, 86 - burn prohibited in CV.

(type = character*16, default = none, units = none)

(4) CDIM

Characteristic dimension of control volume.

If -1 or DEF is input, then CDIM is the radius of a sphere $\{3V/(4\pi)\}^{1/3}$ with volume equivalent to the total volume of the control volume.

(type = real/integer/character*3, default = -1, units = m)

(5) TFRAC

Time fraction of burn before propagation is allowed. It must satisfy $0.0 \leq \text{TFRAC} \leq 1$.

(type = real, default = 0.0, units = none)

(6) CDDH

Value of CDIM during DCH.

(type = real, default = CDIM, units = m)

(7) TFDH

Value of TFRAC during DCH.

(type = real, default = TFRAC, units = none)

Example

```
BUR_BRT 1 ! N  CV NAME  KEY  CF NAME  CDIM  TFRAC  CDDH  TFDH
            1 CV10       CF    CF89   3.46   0.0   3.46   0.0
```

BUR_CC – Combustion Completeness Option

Optional

The parameters determining the option used to calculate the combustion completeness for each volume are on this record.

(1) NUMCC

Number of user-input data strings used to calculate the combustion completeness for each volume, dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMCC. The first three entries are required on each table line if this record is present.

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) ICVNUM

User control volume name.

If -1 is entered, then this combustion completeness specification will be used in all the volumes except those redefined on subsequent table rows.

(type = character*16/integer, default = none, units = none)

(3) ICCFLG

Option to use for combustion completeness.

(a) -1 or CF

Calculate combustion completeness from control function (3a) CFNAME.

(b) 0 or SC
Use correlation (sensitivity coefficient C2202).

(c) 1 or CONST
Use constant value for completeness. Input the constant as entry (3b) on this row.

(type = integer / character*5, default = 0, units = none)

optionally followed by

(3a) CFNAME
Control function name from which combustion completeness is calculated. Required if and only if ICCFLG = -1 or CF.

(type = character*16, default = none, units = none)

or

(3b) CC
Constant value of combustion completeness to use (must satisfy $0.0 \leq CC \leq 1.0$). Required if and only if ICCFLG = 1 or CONST.

(type = real, default = none, units = none)

Otherwise,

(4) ICCDCH
Value of ICCFLG, during DCH.

Optional field.

(a) -1 or CF
Calculate combustion completeness from control function (4a) CFNAME during DCH.

(b) 0 or SC
Use correlation (sensitivity coefficient C2202) during DCH.

(c) 1 or CONST
Use constant value for combustion completeness during DCH. Input the constant as entry (4b) on this row.

(type = integer / character*5, default = ICCFLG, units = none)

optionally followed by

(4a) CFDCH

Control function name from which combustion completeness during DCH is calculated. Required if and only if ICCDCH = -1 or CF.

(type = character*16, default = none, units = none)

or

(4b) CCDH

Value of CC during DCH (must satisfy $0.0 \leq CC \leq 1.0$). Required if ICCDCH = 1 or CONST only.

(type = real, default = CC, units = none)

Example

```
BUR_CC 2  !N  CV NAME  KEY  CFNAME/CC  KEY(DCH)  CFDCH/CCDH
          1   CV10    CF    CF88        CF      CF88
          2   CV20   CONST       0.9   CONST          0.95
```

BUR_FS - Flame Speed Option

Optional

The parameters determining the option used to calculate the flame speed for each volume are on this record.

(1) NUMFS

Number of user-input data strings used to calculate the flame speed for each volume, dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMFS. The first three entries are required if this record is present.

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) ICVNUM

User control volume name. If -1 is entered, then this flame speed specification will be used in all the volumes except those redefined on subsequent table rows.

(type = character*16/integer, default = none, units = none)

(3) IFSFLG

Option specifying how the flame speed is determined.

(a) -1 or 'CF'

Calculate flame speed from control function name (3a) CFNAME.

(b) 0 or 'SC'
Use correlation (sensitivity coefficient C2200).

(c) 1 or 'CONST'
Use constant value for flame speed, one additional field (3b) is required.

(type = integer, default = 0, units = none)

optionally followed by

(3a) CFNAME
Control function name from which flame speed is calculated. Required if and only if IFSFLG = -1 or CF.

(type = character*16, default = none)

or

(3b) FS
Constant value of flame speed to use. Required if and only if IFSFLG = 1 or CONST.

(type = real, default = none, units = m/s)

Otherwise,

(4) IFSDCH
Value of IFSFLG during DCH. Optional field.

(a) -1 or 'CF'
Calculate flame speed from control function (4a) CFDCH during DCH.

(b) 0 or 'SC'
Use correlation (sensitivity coefficient C2200) during DCH.

(c) 1 or 'CONST'
Use constant value for flame speed during DCH, one additional field (4b) is required.

(type = integer, default = IFSFLG, units = none)

optionally followed by

(4a) CFDCH
Control function name from which flame speed during DCH is calculated. Required if and only if IFSDCH = -1 or CF.

(type = character * 16, default = none)

or

(4b) FSDH

Value of flame speed during DCH. Required if and only if IFSDCH = 1 or CONST.

(type = real, default = FS, units = m/s)

Example

```
BUR_FS 1 !N  CV NAME  KEY  CFNAME/FS  KEY(DCH)  CFDCH/FSDCH
          1    CV10    CF   CF83        CF       CF83
```

2.2 MELCOR Input

Records BUR_IGN, BUR_IGN01, BUR_COM, BUR_COM01, BUR_PLT, BUR_BRT, BUR_CC, and BUR_FS may also be included in MELCOR input for any restart. In addition, the initial timestep size at burn initiation can be controlled through the following record.

BUR_TIM – Burn Timestep Information

Optional

The information controlling the system timestep at the initiation of a burn is given on this record. (BUR or other packages may further reduce the size of this initial timestep if necessary and may control the size of subsequent timesteps.)

(1) BURNDT

Burn timestep variable.

(a) < 0.0

The first system timestep during a burn will be ABS(BURNDT) x Total time of burn

(b) = 0.0

First system timestep is 0.2 s

(c) > 0.0

The first system timestep is BURNDT

(type = real, default = 0., units = s)

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require

modification of the FORTRAN source code. The use of sensitivity coefficients is described in Section 7 of the MELCOR Executive (EXEC) Users' Guide.

3.1 BUR Sensitivity Coefficients

C2200 - Flame Speed Parameters

The flame speed in meters per second is evaluated from the equation

$$V = Vbase * Cdil$$

where *Vbase* and *Cdil* are calculated by the following expressions depending on the value of *Y*max, which is the value of y (described below), at the start of the burn and XD, which is the diluent concentration at the start of the burn.

If $0.0 \le Y\max \le Y1$,

$$Vbase = C1 * Y\max + C2$$

If $Y1 < Y\max \le Y2$,

$$Vbase = [C1 + (C2 - C3)/Y1] * Y\max + C3$$

If $Y2 < Y\max \le Y3$,

$$Vbase = [C1 + (C2 - C3)/Y1] * Y\max + C3$$

If $Y3 < Y\max \le Y4$,

$$Vbase = [C1 + (C2 - C3)/Y1 + (C3 - C4)/Y2 + (C4 - C5)/Y3] * Y\max + C5$$

If $Y4 < Y\max \le Y5$,

$$Vbase = [C1 + (C2 - C3)/Y1 + (C3 - C4)/Y2 + (C4 - C5)/Y3 + (C5 - C6)/Y4] * Y\max + C6$$

If $Y5 < Y\max \le 1$,

$$Vbase = [C1 + (C2 - C3)/Y1 + (C3 - C4)/Y2 + (C4 - C5)/Y3 + (C5 - C6)/Y4 + (C6 - C7)/Y5] * Y\max + C7$$

If $0.0 \le Y\max \le Y2$,

$$Cdil = \max\,[C11, C8 + C9 * XD + C10 * XD^2]$$

$$Cdil = \max\,[C11, C8 + C9 * XD + C10 * XD^2] * (Y3 - Y\max)/(Y3 - Y2) + \max\,[0.0, C12 + C13 * XD] * (Y\max - Y2)/(Y3 - Y2)$$

If $Y3 < Y\max \le 1$,

$$Cdil = \max\,[0.0, C12 + C13 * XD]$$

where:

$$Y\max = XH2 + (XH2CC / XCOCC) * XCO$$
$$XD = XH2O + XCO2$$
$$Ci = C2200(i)$$
$$Yi = C2200(13 + i)$$

The values of C2200(i) are sensitivity coefficients as described below:

(1) Constant in base flame speed correlation.
(default = 59.2, units = m/s, equiv = none)

(2) Constant in base flame speed correlation.
(default = 1.792, units = m/s, equiv = none)

(3) Constant in base flame speed correlation.
(default = -9.576, units = m/s, equiv = none)

(4) Constant in base flame speed correlation.
(default = 15., units = m/s, equiv = none)

(5) Constant in base flame speed correlation.
(default = 45., units = m/s, equiv = none)

(6) Constant in base flame speed correlation.
(default = 55., units = m/s, equiv = none)

(7) Constant in base flame speed correlation.
(default = 48.58, units = m/s, equiv = none)

(8) Constant for flame speed diluent effect.
(default = 1.0, units = none, equiv = none)

(9) Constant for flame speed diluent effect.
(default = -4.53, units = none, equiv = none)

(10) Constant for flame speed diluent effect.
(default = 5.37, units = none, equiv = none)

(11) Constant for flame speed diluent effect.
(default = 0.05, units = none, equiv = none)

(12) Constant for flame speed diluent effect.
(default = 1.0, units = none, equiv = none)

(13) Constant for flame speed diluent effect.
(default = -1.29, units = none, equiv = none)

(14) Range limit for flame speed correlation.
(default = 0.1, units = none, equiv = none)

(15) Range limit for flame speed correlation.
(default = 0.2, units = none, equiv = none)

(16) Range limit for flame speed correlation.
(default = 0.3, units = none, equiv = none)

(17) Range limit for flame speed correlation.
(default = 0.4, units = none, equiv = none)

(18) Range limit for flame speed correlation.
(default = 0.6, units = none, equiv = none)

C2201 - Maximum Overshoot of Ignition Limit

A time advancement is vetoed and repeated with a smaller timestep if either the combustible fraction overshoots the ignition limit excessively while flammable or if the diluent or oxygen concentration overshoots the deinerting limit excessively while the combustible concentration is above the ignition limit.

(1) Overshoot allowed on combustible gas concentration.
(default = 0.005, units = none, equiv = TEST1)

(2) Overshoot allowed on deinerting limit (O_2 or CO_2 + H_2O).
(default = 0.01, units = none, equiv = TEST2)

C2202 - Combustion Completeness Parameters

The default combustion completeness (CC) is evaluated as follows:

$$CC = 1 - Y\,\mathrm{min}/Y\,\mathrm{max} \quad (\text{CC constrained such that } 0 \le CC \le 1)$$

where for $Y\mathrm{max} \le$ C2202(2)

$$Y\,\mathrm{min}/Y\,\mathrm{max} = C2202(1)$$

and for $Y\mathrm{max} >$ C2202(2)

$$Y\,\mathrm{min}/Y\,\mathrm{max} = C2202(1) - C2202(3)\,(Y\,\mathrm{max} - C2202(2))$$

*Y*max is LeChatelier's parameter (described above) at the start of the burn, *Y*min is LeChatelier's parameter at the end of the burn, and the values of C2202(i) are sensitivity coefficients as described below.

(1) Slope of combustion completeness correlation for low combustible gas concentrations.
(default = 1.0, units = none, equiv = none)

(2) Combustible gas concentration where slope of combustion completeness correlation changes.
(default = 0.03746, units = none, equiv = none)

(3) Multiplier for combustion completeness correlation at high combustible gas concentration.
(default = 23.4116, units = none, equiv = none)

C2203 - Minimum Debris Temperature to Invoke DCH Parameters

If the temperature of the airborne debris in a control volume falls below this value, then ignition limits, flame speed, and combustion completeness correlations will revert to their non-DCH values.

(1) Minimum airborne debris temperature for DCH parameters.
(default = 600., units = K, equiv = none).

BUR_SC – BUR Sensitivity Coefficients Input Record

Optional

This record allows change of the default BUR sensitivity coefficient values to user-defined input.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the BUR sensitivity coefficient array.
(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
BUR_SC    4  ! N   SC          VALUE       INDEX
               1   2200        0.05        11
               2   2200        0.25        15
               3   2201        0.02         2
               4   2202        25.0         3
```

4 Plot Variables

The BUR package variables that may be used for plot variables are described below. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section. A comparison of MELCOR 2.1 naming conventions with MELCOR 1.8.6 is provided in Appendix 1.

BUR-CPUC	CPU time used by the calculation routines in the BUR package. (units = s)
BUR-CPUE	CPU time used by the edit routines in the BUR package. (units = s)
BUR-CPUR	CPU time used by the restart routines in the BUR package. (units = s)
BUR-CPUT	Total CPU time used by the calculational, edit, and restart routines of the BUR package. (units = s)
BUR-N-SE.n	Number of times a burn has started or ended in control volume index n. It is odd if a burn is occurring in the volume, as well as if no burn is occurring. (units = none)

BUR-O2-RAT.n	The rate of oxygen consumption from burning in Control Volume Hydrodynamics (CVH) volume index n. (units = kg)
BUR-O2-TOT.n	The total amount of oxygen consumed by burning in CVH volume index n. (units = kg)
BUR-H2-RAT.n	The rate of hydrogen consumed from burning in CVH volume index n. (units = kg)
BUR-H2-TOT.n	The total amount of hydrogen consumed by burning in CVH volume index n. (units = kg)
BUR-CO-RAT.n	The rate of carbon monoxide consumption from burning in CVH volume index n. (units = kg/s)
BUR-CO-TOT.n	The total amount of carbon monoxide consumed by burning in CVH volume index n. (units = kg)
BUR-H2O-RAT.n	The rate of steam production from burning in CVH volume index n. (units = kg/s)
BUR-H2O-TOT.n	The total amount of steam produced by burning in CVH volume index n. (units = kg)
BUR-CO2-RAT.n	The rate of carbon dioxide production from burning in CVH volume index n. (units = kg/s)
BUR-CO2-TOT.n	The total amount of carbon dioxide produced by burning in CVH volume index n. (units = kg)
BUR-D2-RAT.n	The rate of deuterium consumed from burning in CVH volume index n. (units = kg/s)

BUR-D2-TOT.n	The total amount of deuterium consumed by burning in CVH volume index n. (units = kg)
BUR-POWER.n	The power generated by burning in CVH volume index n. (units = W)
BUR-ENERGY.n	The energy produced by burning in CVH volume index n. (units = J)
BUR O2 FTOT.m	Total oxygen burned in flow path index m. (units = kg)
BUR H2 FTOT.m	Total hydrogen burned in flow path index m. (units = kg)
BUR D2 FTOT.m	Total deuterium burned in flow path index m. (units = kg)
BUR CO FTOT.m	Total carbon monoxide burned in flow path index m. (units = kg)
BUR H2O FTOT.m	Total steam produced in flow path index m. (units = kg)
BUR CO2 FTOT.m	Total carbon dioxide produced in flow path index m. (units = kg)
BUR FENERGY.m	Total energy produced in flow path index m.

5 Control Function Arguments

The BUR package variables that may be used for control function arguments are described below. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

BUR-CPUC	CPU time used by the calculation routines in the BUR package. (units = s)

BUR-CPUE	CPU time used by the edit routines in the BUR package. (units = s)
BUR-CPUR	CPU time used by the restart routines in the BUR package. (units = s)
BUR-CPUT	Total CPU time used by the calculational, edit, and restart routines of the BUR package. (units = s)
BUR-N-SE(NameCV)	Number of times a burn has started or ended in control volume NameCV. It is odd if a burn is occurring in the volume, and even if no burn is occurring. (units = none)
BUR-LOG	Logical control function argument that is TRUE if any control volume is burning and FALSE otherwise. (type = logical, units = none)
BUR-RAT(NameCV,MAT)	The rate of material MAT production (for production the choices allowed for MAT are H2O – steam, CO2 – carbon dioxide) or consumption (for consumption the choices allowed for MAT are O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide) from burning in CVH volume NameCV. (units = kg/s)
BUR-TOT(NameCV,MAT)	The total amount of material MAT produced (for production the choices allowed for MAT are H2O – steam, CO2 – carbon dioxide) or consumed (for consumption the choices allowed for MAT are O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide) by burning in CVH volume NameCV. (units = kg)
BUR-POWER(NameCV)	The power generated by burning in CVH volume NameCV. (units = W)

BUR-ENERGY(NameCV)	The energy produced by burning in CVH volume NameCV. (units = J)
BUR-FTOT(NameFP,MAT)	Total material MAT produced in flow path NameFP. The choices allowed for MAT are: O2, H2, D2, CO, H2O, CO2. (units = kg)
BUR-FENERGY(NameFP)	Total energy produced in flow path NameFP.

6 Illustrative Input

The illustrative input for the BUR package is a four-volume problem. The inputs for MELGEN and MELCOR are given below. Only the parameters for burn rate, propagation speed, and timestep control are changed from the defaults for this problem.

6.1 MELGEN Input

```
!***Block: Environmental data***
MEG_DIAGFILE ...
MEG_RESTARTFILE ...
...
Program MELGEN
! ***Block: EXEC data***
EXEC_INPUT
EXEC_TITLE ...
EXEC_JOBID ...
...
! ***Block: NCG data***
!
! NONCONDENSIBLE GAS INPUT
!
NCG_INPUT
NCG_ID H2
NCG_ID O2
NCG_ID CO2
NCG_ID CO
!
! CONTROL VOLUME SETUP ...
!
CVH_INPUT
...
! FLOW PATH INPUT ...
!
FL_INPUT
...
! ***Block: BUR data***
! BURN PACKAGE INPUT
!
BUR_INPUT ACTIVE   ! ACTIVATE BURN PACKAGE
!
! INPUT BURN RATE AND PROPAGATION SPEED PARAMETERS
!
BUR_BRT 4 ! N  CV NAME  KEY  CF NAME  CDIM  TFRAC  CDDH  TFDH
            1    CV4   NOTACT          10.    .7
```

```
 2    CV7    ACT           20.   .5
 3    CV8    ACT           25.   .6
 4    CV9   NOTACT         8.    .1
```

6.2 MELCOR Input

```
Program MELCOR
!
! ***Block: MEX (Exec) data***
EXEC_INPUT
EXEC_TITLE ...
EXEC_JOBID ...
! OTHER INPUT
! ***Block: BUR melcor data ***
!
! BURN Timestep INPUT
!
BUR_INPUT
BUR_TIM    -.15
```

7 Discussion of Output

In the regular edits in MELGEN and MELCOR, most of the output is self-explanatory and directly reflects user input. For each MELCOR edit, the volumes that are burning or are inert are summarized. In addition, burning volumes are indicated at every timestep regardless of whether an edit is appropriate. An area that requires further explanation, however, is the node connectivity table for burn propagation. In that table, the orientation is given as -1, 0, or 1. These values represent downward, horizontal, and upward propagation directions and are determined from the user input values of the flow path elevations.

Messages are written to the output file and the special message file for a detonation, the start and end of a deflagration, and for propagation. The ID's are

- BUR0001 - Detonation
- BUR0002 - Deflagration has started
- BUR0003 - Propagation
- BUR0004 - Deflagration has ended

The message lists the time as well as the applicable volumes. These messages are written to the files regardless of whether an edit was desired or not. In addition, when the appropriate input is specified on record BUR_PLT, a plot dump is requested at the same time the message is written out.

8 Error Messages

The error messages in MELGEN pertain to the input and are self-explanatory. The only error messages in MELCOR for the BUR package reflect errors returned from calls to the Control Function package.

Appendix 1

MELCOR 2.1	MELCOR 1.8.6	Comments
BUR-CPUC	BUR-CPUC	CPU time used by the calculation routines in the BUR package.
BUR-CPUE	BUR-CPUE	CPU time used by the edit routines in the BUR package.
BUR-CPUR	BUR-CPUR	CPU time used by the restart routines in the BUR package.
BUR-CPUT	BUR-CPUT	Total CPU time used by the calculational, edit, and restart routines of the BUR package.
BUR-N-SE(NameCV) NameCV – the control volume name.	BUR-N-SE.n n – the control volume index.	Number of times a burn has started or ended in control volume NameCV (n). It is odd if a burn is occurring in the volume, and even if no burn is occurring.
BUR-RAT(NameCV,MAT) NameCV – the control volume name; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	BUR-MAT-RAT.n n – the control volume index; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	The rate of material MAT production or consumption from burning in CVH volume NameCV (n).
BUR-TOT(NameCV,MAT) NameCV – the control volume name; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	BUR-MAT-TOT.n n – the control volume index; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	The total amount of material MAT produced or consumed by burning in CVH volume NameCV (n).
BUR-POWER(NameCV) NameCV – the control volume name.	BUR-POWER.n n – the control volume index.	The power generated by burning in CVH volume NameCV (n).
BUR-ENERGY(NameCV) NameCV – the control volume name.	BUR-ENERGY.n n – the control volume index.	The energy produced by burning in CVH volume NameCV (n).

BUR-FTOT(NameFP,MAT) NameFP – the flow path name; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	BUR-MAT-FTOT.m m – the flow path index; MAT – the material name (O2 – oxygen, H2 – hydrogen, D2 – deuterium, CO – carbon monoxide, H2O – steam, CO2 – carbon dioxide).	Total material MAT produced in flow path NameFP (m).
BUR-FENERGY(NameFP) NameFP – the flow path name.	BUR-FENERGY.m m – the flow path index.	Total energy produced in flow path NameFP (m).

Cavity (CAV) Package
Users' Guide

The Cavity (CAV) package in MELCOR models the attack on the basemat concrete by hot (often molten) core materials. The effects of heat transfer, concrete ablation, cavity shape change, and gas generation are included. These phenomena may be calculated in more than one location (cavity) in a MELCOR calculation, and debris may be transferred between cavities. Specific models are described in more detail, with appropriate references, in the CAV Package Reference Manual. The CAV package has interfaces to the Control Volume Hydrodynamics (CHV) package, the RadioNuclide (RN) package and, through the Transfer Process (TP) package, to the Core (COR) and Fuel Dispersal Interactions (FDI) package.

This users' guide gives a brief introduction to the models in the CAV package, and describes the CAV package input necessary to run MELGEN and MELCOR. Examples are provided. The available control function arguments, plot variables, and sensitivity coefficients are listed and described. Example input decks for running the CAV package in MELCOR are provided, as are brief explanations of the printed output and possible error messages.

Contents

CONTENTS **2**

LIST OF TABLES **3**

1 INTRODUCTION **4**

1.1 Changes since Version 1.8.6 5

2 USER INPUT **6**

2.1 MELGEN Input 6

CAV_DFT – CAV Package Default Scheme Record 6
CAV_ID – Cavity name 8
CAV_DELETE – Ignore all records associated with CAV_ID 8
CAV_CV – Associated Control volume name 9
CAV_Lm – Initial Layer Definition and Layer Contents 9
CAV_C0 – Concrete Declaration 11
CAV_C1 – Concrete Composition 12
CAV_C2 – Other Concrete Properties 14
CAV_DH – Control Functions for Decay Heat 16
CAV_DL – Ablation Delay 17
CAV_G2 – Cavity Shape 18
CAV_G3 – Cavity Shape 19
CAV_GA – Additional Geometry Points 20
CAV_Ra – Rupture/Overflow Input 21
CAV_SP – Definition of Parametric Debris Spreading 22
CAV_TP – *Out* Transfer Process Name 24
CAV_U – Miscellaneous Control and Model Parameters 25

2.2 MELCOR Input 32

CAV_ID – Cavity name 33
CAV_C0 – Concrete Declaration 33
CAV_C1 – Concrete Composition 33
CAV_C2 – Other Concrete Properties 33
CAV_G1 – Parameters of the CORCON Coordinate System 33
CAV_G2 – Cavity Shape 33
CAV_G3 – Cavity Shape 33
CAV_GA – Additional Geometry Points 33
CAV_Ra – Rupture/Overflow Input 33
CAV_U – Miscellaneous Model and Control Parameters 34

3 SENSITIVITY COEFFICIENTS **34**

2300 – Ablation Enthalpy 34
2302 – Melt Viscosity Enhancement 35

2303 – Internal Debris Spreading Model Coefficient 35
2306 – Heat Transfer, Above Melt..... 35
2309 – Heat Transfer, Layer Bulk to Interface..... 36
2312 - Iteration Method Selector for CCSURF 37
2315 - Melt Eruption and Water Ingress Parameters..... 37
CAV_SC – CAV Sensitivity Coefficients Input Record 38

4 PLOT VARIABLES.....39

5 CONTROL FUNCTION ARGUMENTS 41

6 EXAMPLE INPUT.....44

6.1 Example for Normal Cavity Input..... 44

6.2 Stand-Alone Cavity Example..... 46

7 DISCUSSION OF OUTPUT.....47

8 DIAGNOSTIC AND ERROR MESSAGES 48

APPENDIX 1.....50

List of Tables

Table 2.1 Default Concrete Compositions (in mass %)..... 12
Table 2.2 Table 2.2 Default Concrete Properties 15

1 Introduction

The MELCOR Cavity (CAV) package is used to model the interactions between core debris and concrete in one or more locations in a MELCOR calculation; modeling is based on the CORCON-Mod3 code. The capabilities of the CAV package are briefly summarized in this section. More details and references are provided in the CAV Package Reference Manual.

The physical system considered by the CAV package consists of an axisymmetric concrete cavity containing debris in one or more layers. The package calculates heat transfer rates from the debris to the concrete and to the top surface of the debris, as well as the heat transfer between layers. After calculating the heat transfer rates, the concrete ablation rate is determined and the ablation products are added to the cavity contents. Chemical reactions involving gaseous products of concrete decomposition (H_2O and CO_2) with the material in the cavity are calculated and the products are transported to the appropriate layer.

Boundary conditions for the top surface of the debris are obtained from an associated control volume in the Control Volume Hydrodynamics (CVH) package, which also serves as a sink for heat and gases released during the interaction. If there is a water pool in the associated volume, it is assumed to overlie the debris; heat transfer to it is calculated using the same full boiling curve employed in CORCON. (The effects of the introduction of gas bubbles at the lower surface of the pool—referred to as barbotage—and of subcooling were added in CORCON-Mod3. These enhancements may be disabled, if desired.)

By default, the CAV package considers all debris, metallic and oxidic, to be mixed into a single layer. However, the user may choose modeling that considers multiple layers. Two options are available. The first, equivalent to modeling in CORCON-Mod2, does not permit mixing of metals and oxides. It allows a maximum of three layers: metal, heavy oxide below the metal, and light oxide above the metal. The second option invokes mechanistic calculations that consider the possibility of mixing heavy oxides into the metals and/or metals into the light oxides by gas flows at their mutual interfaces (in competition with reseparation under gravity).

The user may choose to specify the initial contents of one metal layer and/or one oxide layer, or of a single mixed-phase layer. If both pure layers are specified, their initial orientation will be determined by their calculated densities, with the densest on the bottom and the lightest on top. Subsequent behavior will be determined by the mixing option in effect.

In most cases, however, no initial contents will be specified; debris will be deposited into the cavity from the Core (COR) package or the Fuel Dispersal Interactions (FDI) package, through the Transfer Process (TP) package. The user may also specify deposition rates directly by using the External Data File (EDF) package and the TP

package. In either case, relative densities and mixing options will determine the layering of the deposited debris.

The decay heat for the cavity can be obtained directly from a user-input control function or calculated using the Decay Heat (DCH) and RadioNuclide (RN) packages. In the former case, or if the RN package is not active to track fission-product relocation, two additional user-input control functions must be specified to define the fraction of total decay heat appearing in each phase (oxide and metal). The user may specify these functions in any case, overriding the MELCOR-calculated results. The simple decay heat model from stand-alone CORCON is not available.

If the RN package is active, it will track the radionuclide inventories associated with the debris. Any initial inventory must be specified by input to the RN package on the RN1_FPN tabular record. When material is transferred into a cavity by the TP package, a parallel transfer of radionuclides is made, also by the TP package. This requires specific input to the TP package, as will be discussed later.

If there is more than one cavity, material may be transferred between cavities. If the RN package is active, the radionuclide inventory associated with the transferred material is also relocated. In this case, the TP package is not involved.

Transfer of material between cavities can be initiated under three types of conditions: axial rupture, radial rupture, or a *TRUE* value returned from a control function. Each of the three types of ruptures (axial, radial, and control function) can result in overflow to another cavity, but only one-way transfers are allowed between cavities (e.g., cavity 1 can overflow to cavity 2, which then overflows to cavity 3, but cavity 3 cannot overflow back to cavity 1 or 2). Rather than specifying an overflow cavity, if desired, the user can either 1) force the calculation to be stopped when rupture is predicted, or 2) specify that any ruptures will be ignored and the calculation continued (with no material transferred out of the cavity) as if additional concrete still remained.

A cavity is considered to have ruptured axially if the basemat is breached and to have ruptured radially if the outer wall is breached. Any material above the elevation at which the rupture occurs will be transferred to a second (user-defined) cavity. The rupture elevation for axial rupture is that of the lowest ablated point, while that for radial rupture is the elevation with minimum remaining wall thickness. Alternatively, the user can control the occurrence of a rupture and the corresponding elevation with control functions. If more than one of the three rupture conditions occurs, the one corresponding to the lowest elevation is chosen.

1.1 Changes since Version 1.8.6

In most cases, the CORCON model in the CAV package of MELCOR 1.8.5 did not predict the quenching of core debris in the cavity in the presence of large amounts of water. This allowed the erosion of concrete, the production of noncondensible and flammable gasses by oxidation of metals in the debris, and the release of fission products to continue unabated.

Although the coefficient for heat transfer between debris and water can be increased through use of the "Miscellaneous Control and Model Parameters" input with keyword BOILING, this usually has little effect because heat transfer within the debris imposes another severe limit on heat removal. The assumption in CORCON is that water cannot break up or penetrate debris, so that heat can only be removed from frozen debris by conduction to the surface. The result is that (for typical compositions and internal heating rates) debris beds thicker than a few inches cannot be kept frozen.

Starting in MELCOR 1.8.6, the user may promote quenching by increasing the values of the thermal conductivities used in the heat transfer calculation. We emphasize that this is a parametric control, not a model.

New plot variables have been made available to facilitate investigation of crust formation and heat transfer.

2 User Input

The user input for the CAV package is described in this section. In MELGEN, one complete set of records is required for each cavity, except for CAV_DFT input record. An arbitrary number of cavities may be defined. In MELCOR, certain properties of an inactive cavity (one which does not yet contain material) may be altered, but the number of cavities defined in the problem cannot be changed.

2.1 MELGEN Input

CAV_DFT – CAV Package Default Scheme Record

Optional

This record modifies the default set of values in the CAV package to the "1.8.6 standards". Note that these values may not reflect current MELCOR best practice modeling parameters for the analysis of severe accidents. If the record is present, one floating point field must be present. This record can be specified once and is not needed for each cavity (i.e. the record is not associated with any particular cavity input accompanied by CAV_ID).

(1) DEFAULT

Mode of default scheme. Permitted values are '1.86' and '2.0'.

(type = real, default = 2.0, units = none)

Example

```
CAV_DFT  1.86
```

If the DEFAULT field is 2.0, no change is made to the existing default values. If the DEFAULT field is 1.86, the following CAV package default values will be modified, unless otherwise specified by the user input record.

Record	DEFAULT = 1.86	DEFAULT = 2.0
EMISS.OX	0.6	0.9
EMISS.MET	0.6	0.9
EMISS.SUR	0.6	0.9
BOILING	None	10.0
COND.OX	1.0	5.0
COND.MET	1.0	5.0

A complete set of the following records (at least the required one) must be input for each cavity. They define:

(1) The initial cavity size, shape, concrete type, and contents (if any);
(2) The control volume that provides boundary conditions and the transfer process (if any) that will deposit material into the cavity;
(3) The method for calculating internal (decay) heating, (if not the default); and
(4) Miscellaneous control and model parameters;

although not necessarily in that order.

The following input records are required to specify cavity:

CAV_ID – Cavity name

Required

This record defines a name for the cavity and optionally a user-defined sequence number.

(1) CAVNAM

User-defined cavity name.

(type = character*16, default = none, units = none)

Example

```
CAV_ID  CAV1  1
```

CAV_DELETE – Ignore all records associated with CAV_ID

Optional

This record is processed after all records in the input file are read. It will turn off all CAV records related to the CAVNAM identifier.

(1) CAVNAM

User-defined cavity name.

(type = character*16, default = none, units = none)

Example

```
CAV_DELETE   CAV1
```

CAV_CV – Associated Control volume name

Required

This record identifies the control volume associated with each cavity.

(1) ICAVN

User name of associated control volume

(type = integer, default = none, units = none)

Example

```
CAV_CV CVolume732       ! Associated control volume name
```

CAV_Lm – Initial Layer Definition and Layer Contents

$1 \leq m \leq 9$, m is the layer number, used for input only
Optional

This record can be used to specify that material will be present in cavity CAVNAM at the start of a calculation. Normally, material is not present in the cavity at the start of a calculation, but is introduced into the cavity during the transient from the other packages of MELCOR.

If it is desired to begin a calculation with material in cavity CAVNAM, the initial temperature for each layer in the cavity must be entered on a CAV_L*m* record. One layer containing only metals and/or one layer containing only oxides, or a single layer containing both metals and oxides may be defined. No use is made of the actual value or values input for *m*; the ordering of the layers, if there are two, will be determined by their calculated densities.

If elimination of a layer that was defined in a previous version of the input deck is desired, this record can also be used to delete it. (Note that if the replacement feature of the input processor is used, only the last CAV_L*m* record in the input stream will be used.) Any CAV_L*m* records with the same *m* will then be ignored.

These records specify the initial layer contents. There can be up to 9 of them, with each record consisting of a head tabular record and then the tabular lines follow. The head record consists of three fields: the first entry is a character or integer key,

determining if the user wants to define a layer or delete it; the second entry is a real variable that specifies the initial temperature of the layer; the third entry is an integer variable which specifies the number of species in the given layer. Then the tabular lines follow. Each tabular line consists of exactly three fields. Multiple data triplets cannot be specified on a record, each triplet should be input on a separate line. The first entry on each tabular line is an integer line identifier; the second entry is a character variable identifying a species, and the third entry is a real variable that specifies the initial mass of the species.

(1) IKEY

Flag determining if user wants to define a layer or delete. Flag takes on the following values:

(a) 1 or 'DELETE'
Deleting a layer.

(b) 2 or 'DEFINE'
Defining a layer.

(type = integer / character*8, default = none, units = none)

The following fields are required if, and only if, IKEY = 2 or IKEY = DEFINE:

(2) TEMP

Initial temperature of layer.

(type = real, default = none, units = K)

(3) NUMMAT

Dimension of the masses table.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMMAT:

(1) N

User-input number for the table's row.

(type = integer, default = none, units = none)

(2) MATNAM

Keyword identifying species. Possible values include AL2O3, CAO, CR2O3, FEO, NIO, SIO2, UO2, ZRO2, AL, C-C, CR, FE, NI, U, and ZR. Other species listed in Appendix A of the CAV Package Reference Manual may also be used.

(type = character*8, default = none, units = none)

(3) SPMASS

Initial mass of material.

(type = real, default = none, units = kg)

Example

```
CAV_L1   DEFINE  1350. 1 ! NUMBER MATNAM          MASS
                             1      UO2            3006.
CAV_L2   DEFINE  1250. 2 ! NUMBER MATNAM          MASS
                               1     FE            4705.56
                               2     ZR             914.6
```

CAV_C0 – Concrete Declaration

Required

This record specifies the type of concrete in the cavity.

(1) ICON

Type of concrete in cavity.

(a) 1 or 'NONSTANDARD',
One additional field is required (2a).

(b) 2 or 'STANDARD',
One additional field is required (2b).

(type = integer / character*16, default = none, units = none)

If ICON = 1 or NONSTANDARD, then

(2a) NonStandardCONTYP

Name of nonstandard concrete type.

(type = character*16, default = none, units = none)

If ICON = 2 or STANDARD, then

(2b) StandardCONTYP

Type of concrete in cavity.

(a) 1 or 'CORCON BASALT,',
Basaltic aggregate concrete.

(b) 2 or 'CORCON L/CS',
Limestone aggregate/common sand concrete.

(c) 3 or 'CORCON GSEUS,
Generic SE United States concrete.

(d) 4 or 'CORCON SRP',
Savanna River Site Concrete

(e) 5 or 'BASALT,
Basaltic concrete.

(f) 6 or 'LIMESTONE/CS',

Limestone/common sand concrete.

(g) 7 or 'CRBR',
Clinch River Breeder Reactor concrete.

(type = integer / character*16, default = none, units = none)

Example

```
CAV_C0 'NONSTANDARD'    'USER-INPUT' ! Concrete declaration
```

CAV_C1 – Concrete Composition

Required if ICON on the CAV_C0 record is NONSTANDARD only, cannot be set for ICON = STANDARD

The data are entered in a table. Each tabular line consists of an integer line identifier, a character keyword, and a real number. The keyword identifies the species, and the real number specifies the species mass fraction in the concrete. Reinforcing bars are considered part of the concrete composition in MELCOR, unlike stand-alone CORCON. The default compositions for standard concretes contain no rebar; therefore, the rebar content should be included as part of these data, even for standard concretes. The composition is not limited to iron. This feature can be used to model metal inserts in the concrete in some experiments. These records specify the concrete composition. Unnormalized mass fractions may be entered. If a standard concrete type was specified for ICON on the CAV_C0 record, its default composition is given in Table 2.1.

Table 2.1 Default Concrete Compositions (in mass %)

SPECIES	CORCON			BASALT	LIMESTONE/CS	CRBR
	1[1]	2[2]	3[3]			
SIO2	54.84	35.80	3.60	65.00	36.00	4.00
TIO2	1.05	0.18	0.12	.	.	.
MNO	.	0.03	0.01	.	.	.
MGO	6.16	0.48	5.67	.	.	.
CAO	8.82	31.30	45.40	.	.	.
NA2O	1.80	0.082	0.078	.	.	.
K2O	5.39	1.22	0.68	.	.	.
FE2O3	6.26	1.44	1.20	.	.	.
AL2O3	8.32	3.60	1.60	20.00	5.00	3.00

CR2O3	.	0.014	0.004	.	.	.
CO2	1.50	21.154	35.698	.	.	.
H2OCHEM	2.00	2.00	2.00	.	.	.
CACO3	.	.	.	3.00	48.00	81.00
CA-OH-2	.	.	.	8.00	8.00	8.00
H2OEVAP	3.86	2.70	3.94	4.00	3.00	4.00

[1]CORCON Basaltic aggregate concrete
[2]CORCON Limestone aggregate/common sand concrete
[3]CORCON Generic SE United States concrete

(1) LENTAB

Dimension of the mass fractions table.

(type = integer, default = none, units = none)

The following data are input as a table with length LENTAB:

(1) N

User-input number for table's row.

(type = integer, default = none, units = none)

(2) MATNAME

Keyword identifying species. Possible values include AL2O3, CAO, CR2O3, FEO, NIO, SIO2, UO2, ZRO2, AL, C-C, CR, FE, NI, U, and ZR. Other species listed in Appendix A of the CAV Package Reference Manual may also be used

(type = character, default = none, units = none)

(3) XMFRCT

Mass fraction of material.

(type = real, default = none, units = dimensionless)

Example

```
CAV_C1   6 ! NUMBER MATNAME     MASS FRACTION
               1      SIO2          0.493
               2      AL2O3         0.035
               3      FE            0.119
               4      H2OEVAP       0.053
               5      CACO3         0.211
               6      CA(OH)2       0.088
```

CAV_C2 – Other Concrete Properties

Required if ICON on the CAV_C0 record is *NONSTANDARD* composition only, optional otherwise

All records are required if ICON is not a standard composition. If concrete is standard and if a record appears, all six numbers should be entered. Defaults for standard concretes are given in Table 2.2. Default values are available for the standard concrete types (values are listed below), but these defaults should not be considered as recommended values.

Table 2.2 Table 2.2 Default Concrete Properties

PROPERTY	CORCON			BASALT	LIMESTONE/CS	CRBR
	1[1]	2[2]	3[3]			
DENSCT (kg/m^3)	2340.	2340.	2340.	2400.	2340.	2340.
TSOLCT (K)	1350.	1420.	1690.	1350.	1420.	1690.
TLIQCT (K)	1650.	1670.	1875.	1650.	1670.	1875.
TABLCT (K)	1450.	1500.	1750.	1450.	1500.	1750.

[1]CORCON Basaltic aggregate concrete
[2]CORCON Limestone aggregate/common sand concrete
[3]CORCON Generic SE United States concrete

(1) DENSCT

Density of concrete.

(type = real, units = kg/m^3, default = Table 2.2 for standard concrete, none for other concrete)

(2) TSOLCT

Solidus temperature of concrete.

(type = real, units = K, default = Table 2.2 for standard concrete, none for other concrete)

(3) TLIQCT

Liquidus temperature of concrete

(type = real, units = K, default = Table 2.2 for standard concrete, none for other concrete)

(4) TABLCT

Ablation temperature of concrete.

(type = real, units = K, default = Table 2.2 for standard concrete, none for other concrete)

(5) TINCT

Initial temperature of concrete.

(type = real, default = 298.0, units = K)

(6) EMISCT

Emissivity of concrete.

(type = real, default = 0.6, units = none)

Example

```
CAV_C2 2400. 1420. 1670. 1573. 374. 0.6 ! Concrete properties
```

CAV_DH – Control Functions for Decay Heat

Optional

By default, the decay heat power input to the cavity is determined by MELCOR using the inventory of radionuclides in the cavity (calculated by the RN package) and their specific decay powers (defined by the DCH package). If the RN package is inactive (as in analysis of a simulation experiment) or if the user wishes to override or modify its results, control functions may be specified on this record, which will then be used in calculation of the decay heat. This provides the capability to define a total decay heat other than that calculated by MELCOR and/or to define the partition of heat input among layers in the pool.

Unless IPDHCF = 1 or IPDHCF = 'UseDCH' and the RN package is active, both IPOXCF and IPMCF must be specified = -1 or 'CF' as MELCOR has no other way to determine the partition of heating between layers.

(1) IPDHCF

Flag for total decay heat

(a) 0 or 'NODECAYHEAT'
Set decay heat to zero.

(b) 1 or 'UseDCH',
Use calculation from decay heat and radionuclide packages.

(c) -1 or 'CF'
Use control function which defines the total decay heat in the cavity.

(type = integer / character*16, default = none, units = none)

If IPDHCF = -1 or CF, this field must be input:

(2) NameCF_IPDHCF

Control function name.

(type = character*16, default = none, units = none)

If IPDHCF = 1 or UseDCH, or IPDHCF = -1 or CF, then

(3) IPOXCF

Flag for fraction of heat in oxide phase.

(a) 0 or 'USEMELCORSPLIT'
Use split calculated by MELCOR (only available when IPDHCF =1 or UseDCH)

(b) -1 or 'CF'
Control function specifies fraction of total decay heat to be put into the oxide phase

(type = integer / character*16, default = none, units = none)

If IPOXCF = -1 or CF, this field must be input:

(4) NameCF_IPOXCF
Control function name.

(type = character*16, default = none, units = none)

If IPDHCF = 1 or UseDCH, or IPDHCF = -1 or CF, then

(5) IPMCF
Flag for fraction of heat in metal phase.

(a) 0 = 'USEMELCORSPLIT'
Use split calculated by MELCOR (only available when IPDHCF =1 or UseDCH).

(b) -1 = 'CF'
Control function specifies fraction of total decay heat to be put into the metal phase.

(type = integer / character*16, default = none, units = none)

If IPMCF = -1 or CF, this field must be input:

(6) NameCF_IPMCF
Control function name.

(type = character*16, default = none, units = none)

Example

```
CAV_DH CF CF50 CF CF53 CF CF52 ! CFs for decay heat
```

CAV_DL – Ablation Delay

Optional

This record may be used to delay the start of concrete ablation until a user-defined criterion is met in addition to the requirements of the CORCON model. It may be used to model such phenomena as the burn-through of a steel liner. If this record is present, the possibility of ablation will not be considered until the specified logical control function becomes .TRUE. If the control function is already .TRUE. in MELGEN, a warning will be issued. If, during MELCOR time advancement, the function is .TRUE. and there is material in the cavity, ablation will be considered for

that time step and all subsequent timesteps. A latch is set within the CAV package so that further changes in the control function will be ignored.

(1) IPDEL

Name of a logical control function to define an additional criterion that must be met before ablation will be calculated.

(type = character*16, default = none, units = none)

Example

```
CAV_DL   CF335
```

Note: The only option available at this time for the initial geometry of the concrete cavity is the "flat-bottom cylinder" option of CORCON. Note that CORCON uses its own coordinate system, with z positive down, which is currently totally independent of the rest of MELCOR.

CAV_G1 – Parameters of the CORCON Coordinate System

Required

(1) NRAYS

Number of rays in the CORCON system (≤100).

(type = integer , default = none, units = none)

(2) ZO

Axial coordinate of center of the ray system. This value should be chosen such that it will be central to the eroded cavity, so that the angles between rays and the surface are not too acute.

(type = real, default = none, units = m)

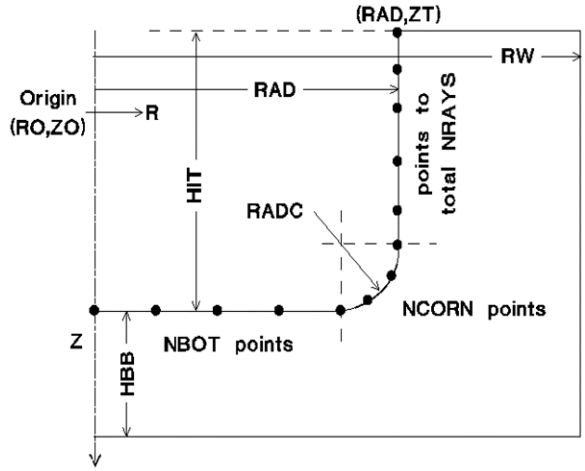
Example

```
CAV_G1 50 2. ! Parameters of CORCON coordinate system
```

CAV_G2 – Cavity Shape

Required

This record specifies the initial cavity shape. The associated variables are illustrated in the figure below, where NBOT=5, NCORN=2, and NRAYS=13. Experience has shown that use of closely spaced points on the corner may lead to calculational difficulties. We therefore suggest that small values of RADC and/or large values of NCORN be avoided.



(1) ZT

Z coordinate of the cylinder top edge.

(type = real, default = none, units = m)

(2) RAD

Radius of the cylindrical cavity.

(type = real, default = none, units = m)

(3) HIT

Height of the cylindrical cavity.

(type = real, default = none, units = m)

(4) RADC

Radius of the corner (transition from cylindrical wall to floor) of the cavity.

(type = real, default = none, units = m)

Example

```
CAV_G2    1.9    0.85    3.1    0.1 ! Cavity shape
```

CAV_G3 – Cavity Shape

Required

This record continues specification of the initial cavity shape.

(1) FLAG_RW

If FLAG_RW = 'TF', then tabular function defines external radius of concrete as a function of *z*, else RW defines the external radius of concrete.

(a) 1 or 'TF',
Use tabular function to define the external radius of concrete as a function of *z*.

(b) 0 or 'VALUE',
Use RW to define the external radius of concrete.

(type = integer / character*8, default = none, units = m)

(2) RW

Name of tabular function or external radius of concrete. The interpretation is dependent on the value of FLAG_RW.

(type = character*16 / real, default = none, units = none)

(3) HBB

Thickness of concrete below the bottom of the cavity.

(type = real, default = none, units = m)

(4) NBOT

Number of points defining the flat bottom of the cavity, ≥ 2.

(type = integer, default = none, units = none)

(5) NCORN

Number of points defining corner.

(type = real, default = none, units = none)

Example

```
CAV_G3    VALUE    1.01    0.2    10    4 ! Cavity shape
```

CAV_GA – Additional Geometry Points

Optional

This record is used to add one or more points to the geometry description above the top of the cavity, as previously specified, with decreasing z coordinates in the upward direction. The total number of points may not exceed the maximum of 100. If multiple points are specified, each must define a point lying above all previous points.

(1) NUMPOINT

Number of triplets that are used to add one or more points to the geometry description above the top of the cavity; dimension of the following table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMPOINT:

(1) N

User-input number for the table's row.

(type = integer, default = none, units = none)

(2) NEXTRA

Number of points to be added. If NEXTRA > 1, the points will be uniformly spaced along a line segment from the old last point to the new one.

(type = integer, default = none, units = none)

(3) REXTRA

R-coordinate of the new last point.

(type = real, default = none, units = m)

(4) ZEXTRA

Z-coordinate of the new last point.

(type = real, default = none, units = m)

Example

```
CAV_GA  3 ! N   POINTS TO ADD       R             Z
            1         1           0.855         -8.0
            2         2           0.91          -8.2
            3         4           0.94          -8.5
```

CAV_Ra – Rupture/Overflow Input

a = R, A, or T
Optional

This record identifies the cavity to which material will be moved if the present cavity ruptures, and optionally identifies control functions to control when a rupture will occur and the elevation of the rupture. Any combination of the three different types of rupture may be specified: axial (a = A), radial (a = R), or control function (a = T). The tests for each type of rupture are discussed in Section 1.

(1) NOVC

Options.

(a) -1 or 'STOP',

Stop the calculation if this type of rupture occurs.

(b) -2 or 'CONTINUE',

Calculation should continue unchanged (as if infinite concrete were available), even if rupture is predicted.

(c) 1 or 'CAVITY',

The cavity will receive overflowing material following a rupture.

(type = integer / character*8, default = -1, units = none)

If NOVC = 1 or CAVITY, then the following fields are required:

(2) NAMECAV

Cavity name.

(type = character*16, default = none, units = none)

(3) NCFRUP

Name of the logical control function that will trigger rupture of this cavity. This input is optional for a = A or R, and required for a = T (see Section 1 for further details).

(type = character*16, default = none, units = none)

(4) NCFREL

Name of the real-valued control function that specifies the rupture elevation for this cavity. Note that the elevation must be given in the CORCON coordinate system (positive downward). This input is optional for a = A or R, and required for a = T (see Section 1 for further details).

(type = character*16, default = none, units = none)

Example

```
CAV_RA   CONTINUE
CAV_RR   CONTINUE
CAV_RT   CAVITY   CAV2   CF1   CF8
```

CAV_SP – Definition of Parametric Debris Spreading

Optional

This record may be used to model the spreading of debris in the cavity. MELCOR provides an internal calculated debris-spreading. However, users can define a maximum debris radius as a function of time through a tabular function, control function, or channel of an external data file. The fourth field on the record is optional. If it is absent, heat transfer from the radial surface of the spreading debris will be considered.

(1) SOURCE

Source of data for maximum debris radius as a function of time

(a) 1 or 'TF'

Use data from tabular function.

(b) -1 or 'CF'
Use data from control function.

(c) 2 or 'CHANNELEDF',
Use data from channel of external data file NameCF_TF_EDF.

(d) 0 or 'MODEL',
This option allows the code to internally calculate the debris radius as a function of time. However, this option requires the initial debris radius (RADTINI).

(type = integer / character, default = none, units = none)

If SOURCE = 0, the following record is required:

(2) RADTINI

Initial time-dependent debris radius for the internal model. If a zero value is input, the model will calculate a radius. If a negative value is provided, it is assumed that the initial radius is computed from ABRCH area from the COR package. However, there is a test to ensure that both COR package is active, and the cavity's FLAGTP has a value of 1. If not, a zero value is assigned to RADTINI.

(type = real, default = none, units = m)

If SOURCE = 0, the following is not applicable. It is applicable for SOURCE = 1, -1 and 2.

(2) NameCF_TF_EDF

Name of tabular function, control function, or external data file. The interpretation is dependent on the value of SOURCE.

(type = character*16, default = none, units = none)

If SOURCE = 2 or ChannelEDF, then

(3) NCHAN

Channel number of an external data file

Required if SOURCE = 2 or ChannelEDF only.

(type = integer, default = 0, units = none)

(4) HTSIDE

Treatment of heat transfer from radial face of spreading debris. Note: if SOURCE = 1 or 'TF' or -1 or 'CF', this will be the third field. Optional field. If SOURCE = 0, this will be the third field.

(a) 1 = 'ADIABATIC'
Heat transfer suppressed.

(b) 2 = 'HEATTRANSFER',
Heat transfer from the radial surface of the spreading debris will be considered.

(type = integer / character*16, default = 2, units = none)

Example

```
CAV_SP   CF   CF49   HeatTransfer
```

Or

CAV_SP MODEL 0.375 HeatTransfer

CAV_TP – *Out* Transfer Process Name

Optional

This record identifies the *out* transfer process name for transferring mass from the COR or FDI packages to the cavity. If absent, no mass will be transferred into the cavity from the COR or FDI packages.

(1) NTPOT

Out transfer process name (this must be one of the processes listed on TP_OUT input record) associated with the *in* transfer process for masses and energies from the COR or FDI package.

(type = character*16, default = none, units = none)

Note that this requires input to the TP package. The appropriate input on the TP_OUT record is NMSOT = 5, NPOTOI = in process name on either the COR_TP record or the FDI_LOC record as appropriate. For a transfer from FDI, a unity translation matrix should be used (OUTMTX=DEF1). For a transfer direct from COR, a user-defined matrix should be used (OUTMTX = 'user-defined name' of the translation matrix), the corresponding TP_MTX record should define NROW = 5 and NCOL = 6 (the CAV package does nothing with control poison from the core), and matrix elements 1/1, 2/2, 3/3, 4/4, and 5/5 equal to 1.0.

(2) RNTPOT

Out transfer process name for associated radionuclides (in addition to the NTPOT above). This field is required if RN package is active, otherwise, it must be absent.

(type = character*16, default = none, units = none)

In order to transfer the radionuclides along with the total masses, additional TP names must be defined. These radionuclide TP names must be defined in addition to the corresponding TP names defined for total mass and energy transfers by the COR or FDI, and CAV input, otherwise an error message will be issued. See the TP Package Users' Guide for more details.

Example

```
CAV_TP   TP101  TP601
```

CAV_U – Miscellaneous Control and Model Parameters

Optional

(1) NUMMISC

Number of user-input miscellaneous control and model parameters, dimension of the table below.

(type = integer, default = 0, units = none)

The following fields define the table. The number of rows must be equal to NUMMISC. Each row consists of integer number N, character variable (keyword) and either a real or integer variable. The keyword identifies the parameter and the real or integer variable specifies the value of the parameter.

(1) N

User-input number of table's row

(type = integer, default = none, units = none)

The character variables are:

(2) BOILING

Treatment of enhancements to the boiling curve for heat transfer to overlying coolant. A multiplier may be applied to the standard curve (including subcooling and gas barbotage enhancements), or the enhancements may be suppressed

(3) ICHT

Option with one of the following values

(a) 0 or 'MOD3'

Use standard CORCON-Mod3 model, including enhancements

(b) -1 or 'VALUE'

Use value as multiplier on standard model

(c) 2 or 'MOD2'

Suppress enhancements (the CORCON-Mod2 model)

(type = integer / character, default = 0, units = none)

The following field is required if ICHT = -1 or 'VALUE', or ICHT = 2 or 'MOD2':

(4) CHT

Multiplier applied to 'CORCON-Mod3' boiling or 'CORCON-Mod2' boiling

(type = real, default = none, units = none)

(2) COKE

Coking flag

(3) ICOK

Option with one of the following values

(a) 1 or 'ENABLE'

This value will enable production of condensed carbon during oxidation of zirconium

(b) 0 or 'SUP'

This value will suppress the reaction

(type = integer / character, default = 0, units = none)

(2) CTOXYREA

Treatment of chemical reactions involving concrete decomposition products. Zero value will suppress the reduction of ablated oxides by debris metals. If suppressed by previous input, a value of one will enable the reaction

(3) ICHEM

Option with one of the following values

(a) 1or 'INCLUDE'

Include oxides and gases as reactants

(b) 0 or 'EXCLUDE'

Exclude oxides, and consider gases only (the CORCON-Mod2 model)

(type = integer/ character, default = 1, units = none)

(2) EMISS.OX

Emissivity of the oxide phase

(3) EO

Value of emissivity of the oxide phase

(type = real, default = 0.6, units = none)

(2) EMISS.MET

Emissivity of the metal phase

(3) EM

Value of emissivity of the metal phase

(type = real, default = 0.6, units = none)

(2) EMISS.SUR

Emissivity of the surroundings

(3) ES

Value emissivity of the surroundings

(type = real, default = 0.6, units = none)

(2) GFILMBOTT

Selection of gas or slag film model for the melt/concrete interface at the bottom surface of the debris. Value of one selects the gas film model

(3) IFILMB

Option with one of the following values

(a) 0 or 'SLAG'
Use slag film model

(b) 1 or 'GAS'
Use gas film model (the CORCON-Mod2 model)

(type = integer / character, default = 1, units = none)

(2) GFILMSIDE

Selection of gas or slag film model for the melt/concrete interface at the radial surface of the debris. Value of one selects the gas film model

(3) IFILMS

The key with one of the following values

(a) 0 or 'SLAG'
Use slag film model

(b) 1 or 'GAS'
Use gas film model (the CORCON-Mod2 model)

(type = integer / character, default = 1, units = none)

(2) HTRBOT

Treatment of debris-to-surface heat transfer at the bottom surface of the debris. Either a multiplier on the standard model or an alternate analytic form may be specified

(3) IBHTB

Option with one of the following values

(a) 0 or 'STAND'

Use standard CORCON-Mod3 model

(b) -1 or 'MULTIP'
Use multiplier on standard model

(c) 1 or 'SENSC'
Use alternate form of correlation with coefficients defined by sensitivity coefficients C2309(1-3)

(type = integer / character, default = 0., units = none)

Next field is required if and only if IBHTB = -1 or MULTIP

(4) XBM

Use this value as multiplier on standard model

(type = real, default = none., units = none)

(2) HTRINT

Treatment of debris-to-surface heat transfer at interior interfaces between debris layers. Either a multiplier on the standard model or an alternate analytic form may be specified

(3) IIHT

Option with one of the following values

(a) 0 or 'STAND'
Use standard CORCON-Mod3 model

(b) -1 or 'MULTIP'
Use multiplier on standard model

(c) 1 or 'SENSC'
Use alternate form of correlation with coefficients defined by sensitivity coefficients C2309(7-9)

(type = integer / character, default = 0., units = none)

Next field is required if and only if IIHT = -1 or MULTIP

(4) XIM

Use this value as multiplier on the standard model (see SC2309(10).

(type = real, default = none., units = none)

(2) HTRSIDE

Treatment of debris-to-surface heat transfer at the radial surface of the debris. Either a multiplier on the standard model or an alternate analytic form may be specified

(3) IBHTS

Option with one of the following values

(a) 0 or 'STAND'
Use standard CORCON-Mod3 model

(b) -1 or 'MULTIP'
Use value as multiplier on the standard model

(c) 1 or 'SENSC'
Use alternate form of correlation with coefficients defined by sensitivity coefficients C2309(4-6)

(type = integer / character, default = 0., units = none)

Next field is required if and only if IBHTS = -1 or MULTIP

(4) XIM

Use this value as multiplier on the standard model

(type = real, default = none., units = none)

(2) MIXING

Treatment of mixing between metallic and oxidic components of the debris. Option with one of the following values

(3) IMIX

(a) 0 or 'SUP'
Suppress mixing (the CORCON-Mod2 model)

(b) 1 or 'CALC'
Calculate mixing and separation rates from correlations

(c) -1 or 'ENFOR'
Enforce mixing (all debris forms a single mixed layer)

(type = integer / character, default = -1., units = none)

(2) NONIDEAL

Treatment of chemical free energies in the VANESA fission product release model. Because the nonideal oxide model is not operational, options invoking it cannot be used

(3) IDEAL

Option with one of the following values

(a) 1 = 'IDEAL'
Ideal model for both metals and oxides

(b) -1 = 'DIFFER'
Nonideal model for metals, ideal model for oxides

(type = integer, default = 1., units = none)

(2) RADLEN

Path length for the optional aerosol opacity calculation in the calculation of radiative heat loss from the debris surface

(3) RAD

Value of the path length for the optional aerosol opacity calculation

(type = real, default = 0., units = m)

(2) SHAPEPLOT

Inclusion of cavity shape data in the plot file in the form of r and z coordinates of the defining body points. By default these are omitted to reduce the size of the plot file. (see Section 4 for details)

(3) SHAPEPL

The key with one of the following values

(a) 0 or 'EXCLUDE'
Exclude cavity shape data from plot file

(b) 1 or 'INCLUDE'
Include cavity shape data in plot file

(type = integer, default = 0., units = none)

(2) TDEBUG

Time to start CORCON diagnostic print

(3) TDEB

Value of time to start CORCON diagnostic print

(type = real, default = "infinity", units = s)

(2) IBUBX

Optional RN pool scrubbing activation flag

(3) IBUB

Option with one of the following values

(a) 0 or ' NOBUBBLERISE '
RN scrubbing of aerosols and iodine vapor is active

(b) -1 or ' NOSCRUBBINGRN '
All RN scrubbing is inactive

(c) -2 or ' AEROSOLSCRUBBING '
RN scrubbing of aerosols only is active

(d) -3 or ' IODINESCRUBBING '
RN scrubbing of iodine vapor only is active

(type = integer / character, default = 0, units = none)

(2) COND.OX

Optional thermal conductivity of oxidic mixtures, having one of the following meanings

(3) CONDOX

option having one of the following values

(a) 0 or MODEL
Use internal model

(b) 1 or TF
Use table function

(c) 2 or CONST
Constant conductivity

(d) 3 or MULT
Multiplier on conductivity

followed by:

(4) CONDVAL

Name of table if CONDOX = TF
Value of conductivity if CONDOX = CONST, or value of multiplier if CONDOX = MULT.

(2) COND.MET

Optional thermal conductivity of metallic mixtures, having one of the following meanings.

(3) CONDMET

Option having one of the following values

(a) 0 or MODEL
Use internal model

(b) 1 or TF
Use table function

(c) 2 or CONST
Constant conductivity

(d) 3 or MULT
Multiplier on conductivity

followed by:

(4) CONDVAL

Name of table if CONDMET = TF
Value of conductivity if CONDMET = CONST, or value of multiplier if CONDMET = MULT.

(2) WATINGR

Optional switch to use water ingress model.

(3) WIVAL

Option switch having one of the following values

(a) 1 or ON
Use water ingress model

(b) 0 or NONE
Water ingress model is off

(type = integer/character, default = 'NONE', units = none)

(2) ERUPT

Optional switch to use melt eruption model.

(3) MEVAL

Option switch having one of the following values

(a) 1 or ON
Use melt eruption model

(b) 0 or NONE
Melt eruption model is off

(type = integer/character, default = 'NONE', units = none)

Example

```
CAV_U     7  !N  KEYWORD    VALUE(S)
              1  BOILING     -1      1.3
              2  EMISS.OX    0.9
              3  NONIDEAL    -1
              4  RADLEN      5.
              5  SHAPEPLOT   1
              6  COND.OX     CONST  5.
              7  COND.MET    CONST  5.
```

2.2 MELCOR Input

Certain elements of the input for an application of the cavity model may be modified at a restart if the model is not yet active (if the cavity contains no material). These allow changes to the initial geometry, concrete type, and miscellaneous model parameters. This is intended to allow a user to perform several calculations which vary on cavity parameters without the need to rerun the pre-vessel-failure part of the calculation.

In addition, cavity rupture/overflow input can be changed such that a cavity continues to ablate (without any material being transferred out of it) even if a rupture is predicted. This change can be made even if a cavity is already active.

The input is a subset of that described for MELGEN; the permitted records are listed below (any others will be ignored). Details for each record are provided in Section 2.1.

CAV_ID – Cavity name

Required if some other data are changed for the cavity.

This record must be input before all other records. The name must be identical to the one previously input to MELGEN.

CAV_C0 – Concrete Declaration

Optional

If CAV_C0 is present, a complete concrete definition (CAV_C1 and CAV_C2 records) must be included. If CAV_C0 is absent, the other CAV_C* records will be ignored.

CAV_C1 – Concrete Composition

CAV_C2 – Other Concrete Properties

CAV_G1 – Parameters of the CORCON Coordinate System

Optional

If CAV_G1 is present, all of the following records (CAV_G2, CAV_G3, and CAV_GA) must be included, even though the data elements involved may be unchanged from MELGEN. If CAV_G1 is absent, the other CAV_G* records will be ignored.

CAV_G2 – Cavity Shape

CAV_G3 – Cavity Shape

CAV_GA – Additional Geometry Points

This may not be included without CAV_G1, CAV_G2, and CAV_G3, even if these involve no changes from MELGEN.

CAV_Ra – Rupture/Overflow Input

Optional

The only overriding change permitted is to change NOVC to -2 or CONTINUE. NAMECAV, NCFRUP, and NCFREL need not be entered.

CAV_U – Miscellaneous Model and Control Parameters

Optional

Any appropriate combination described in the preceding section may be entered, whether or not present in MELGEN input.

3 Sensitivity Coefficients

The sensitivity coefficients for the CAV package have identifier numbers from 2300 to 2499. Sensitivity coefficient array 2301, recognized by MELCOR 1.8.2, was eliminated in favor of similar capabilities available through a new array, 2309, in MELCOR 1.8.3. Most of the other sensitivity coefficients from previous versions of the CAV package (specifically, 2302, 2304, 2305, part of 2306, 2307, and 2308), have not yet been connected to CORCON-Mod3. This situation will be corrected in later versions of MELCOR.

2300 – Ablation Enthalpy

This coefficient may be used to modify the heat of ablation of concrete. At this time, it has no effect on restart unless the concrete is also redefined.

The heat of ablation is defined herein as the heat required to convert a unit mass of virgin concrete at the user-defined initial temperature, to condensed-phase and gaseous decomposition products, all at the specified ablation temperature. It contains contributions from sensible heat, heat of chemical decomposition, and heat of fusion, and is internally calculated from the CORCON thermochemical database. It is important to realize that all enthalpies are calculated from the same database. Therefore, while increasing the ablation temperature will increase the value calculated for the heat of ablation, it will lead to an equal reduction in the heat necessary to raise the ablation products to a given pool temperature. Thus, the net effect of a change in ablation temperature on the heat necessary to ablate a unit mass of concrete and bring the ablation products to pool temperature is only indirect, through a possible change in the pool temperature predicted. In order to change the energy balance associated with ablation, it is necessary to modify the heat of ablation at a given ablation temperature.

(1) Additional ablation enthalpy to be added to the internally-calculated value.

(default = 0.0, units = J/kg, equiv = DDELH)

2302 – Melt Viscosity Enhancement

This sensitivity coefficient allows to select the current enhancement to the melt viscosity in MELCOR using the Kunitz two-phase multiplier as in Eq(4) or newly added Ramacciotti correlation.

(1) Internal model selection: 0.0=Kunitz correlation, nonzero=Ramacciotti correlation.
(default = 0.0, units = dimensionless, equiv = VISMOD)

(2) Value for Ramacciotti's correlation coefficient 1.
(default = 2.5, units = dimensionless, equiv = RAMC1)

(3) Value for Ramacciotti's correlation coefficient 2. Note that the value between 4.0 to 8.0 is recommended.
(default = 6.0, units = dimensionless, equiv = DDELH)

Note that the selection will affect both oxide and metallic melt.

2303 – Internal Debris Spreading Model Coefficient

This sensitivity coefficient allows to the user to modify as necessary for the integral constant in the internal analytical debris spreading model as SOURCE=MODEL in the CAV_SP input record. A large 2303(1) value implies that the spreading is faster. A smaller value implies that viscosity term dominates. In addition, 2303 is used to model coefficients for the stopping criteria as described. Option 1 assumes the stopping based on a fraction of crust height and crust radius.

(1) Constant for the analytical debris spreading model.
(default = 0.136, units = dimensionless, equiv = RADTCOF)

(2) Stopping option selector. 0 = option 1.
(default = 0.0, units = dimensionless, equiv = RADT_Stop)

(3) For stopping option (1) – minimum fraction of RADT to stop spreading is 0.5.
(default = 0.5, units = dimensionless, equiv = RADT_Stop11)

(4) For stopping option (1) – minimum fraction of solid thickness to stop spreading is 0.5.
(default = 0.5, units = dimensionless, equiv = RADT_Stop12)

2306 – Heat Transfer, Above Melt

These coefficients may be used to modify heat transfer from the surface of the debris pool.

(1) Convective heat transfer coefficient in atmosphere.
(default = 10.0, units = W/m^2K, equiv = HA)

(2) Convective heat transfer coefficient in water pool.
(default = 1000.0, units = W/m^2K, equiv = HCONV)

2309 – Heat Transfer, Layer Bulk to Interface

These coefficients may be used to modify the coefficients in the alternate correlations used for gas-enhanced heat transfer between the interior of a liquid layer (or sublayer) and its surfaces. These alternate correlations are selected by the HTRBOT, HTRINT, and HTRSIDE keywords on CAV_U records. (Note that the model also includes natural convection and conduction limits which are not modifiable through these coefficients.)

There is also a switch to turn off the multiplier HTRINT for the top crust, SC2309(10).

The alternate correlation has the form

$$Nu = A \cdot \mathrm{Re}^B \cdot \mathrm{Pr}^C$$

Here the length scale in the Nusselt and Reynolds numbers is the bubble radius. The velocity in the Reynolds number is the superficial velocity of gas flowing through the layer. The alternate correlation has the form used for bottom and interfacial heat transfer in CORCON-Mod2, and the default values for the coefficients are those used there. Differences in evaluation of materials properties in CORCON-Mod3, however, will result in differences in calculated heat transfer. Also, CORCON-Mod2 used a different correlation for side heat transfer. Although detailed agreement is impossible, the default coefficients A, B, and C have been chosen to give heat transfer coefficients in rough agreement with those in CORCON-Mod2 for a limited number of cases.

(1) Constant A for bubble injection (at the bottom of the lowest layer).
(default = 5.05, units = dimensionless, equiv = CCAHTB)

(2) Exponent B for bubble injection.
(default = 0.5, units = dimensionless, equiv = CCBHTB)

(3) Exponent C for bubble injection.
(default = 0.8, units = dimensionless, equiv = CCCHTB)

(4) Constant A for side heat transfer.
(default = 2.0, units = dimensionless, equiv = CCAHTS)

(5) Exponent B for side heat transfer.

(default = 0.3, units = dimensionless, equiv = CCBHTS)

(6) Exponent C for side heat transfer.
(default = 0.0, units = dimensionless, equiv = CCCHTS)

(7) Constant A for interfaces between layers.
(default = 5.05, units = dimensionless, equiv = CCAI)

(8) Exponent B for interfaces between layers.
(default = 0.5, units = dimensionless, equiv = CCBI)

(9) Exponent C for interfaces between layers.
(default = 0.8, units = dimensionless, equiv = CCCI)

(10) Switch to use/not use HTRINT on top crust (0.0 = do not use, 1.0 = use).
(default = 0.0, units = dimensionless)

2312 - Iteration Method Selector for CCSURF

This SC allows changing the iteration method used in the CCSURF routine. This routine finds the interface temperature between the concrete and the melt using either Newton-Raphson iteration (NR) (the original), the bisection method, or regula falsi. The reason for the optional methods is that the basic Newton-Raphson method, although fastest, is unconstrained and can go outside the allowable bounds during the iteration. The bounds are the concrete ablation temperature and the maximum layer temperature. If the interface temperature is outside of those bounds, then there is no concrete ablation, no heat flux into the concrete, and no solution to the iteration. The other two methods, bisection and regula falsi, are inherently bounded, so they will either find a solution on the interval or can tell that there is no solution. The disadvantage is that they are slower than NR; the slowest is bisection, although this is the surest to converge, followed by regula falsi.

(1) Iteration method selector. Possible values are 1.0 = Newton, 2.0 = bisection, 3.0 = regula falsi.
(default = 3.0, units = dimensionless, equiv = ITMETH)

2315 - Melt Eruption and Water Ingress Parameters

This SC contains model parameters for the melt eruption and water ingress models.

(1) Constant in the Ricou-Spalding correlation.
(default = 0.08, units = dimensionless, equiv = EENT)

(2) Debris diameter.
(default = 0.01, units = meters, equiv = RDEB)

(3) Debris layer void fraction.

(default = 0.3, units = dimensionless, equiv = ALFB)

(4) Minimum conduction zone thickness in water ingress model. If -1.0, minimum is full crust layer thickness, 0.0 means no minimum.

(default = -1.0, units = meters, equiv = DELMIN)

CAV_SC – CAV Sensitivity Coefficients Input Record

Optional

This record allows the user to change the default CAV sensitivity coefficient values to user input ones. This table of records is not required.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the CAV sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only – no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
CAV_SC   4  ! N   SC          VALUE         INDEX
              1   2306        11.0          1
              2   2306        1.5E3         2
              3   2309        2.5           4
              4   2309        0.75          9
```

4 Plot Variables

The CAV package variables that may be used for plot variables are listed and described below. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section. In the following list, *n* refers to the cavity index, lay refers to the desired layer, with HOX referring to the heavy oxide layer, LOX to the light oxide layer, MET to the metal layer, HMX to the heavy mixture layer, and LMX to the light mixture layer. Here heavy and light are in comparison to the metal. Finally gas refers to one of the four gases (H_2, H_2O, CO, or CO_2).

CAV-MTOT.n	Total mass in cavity n. (units = kg)
CAV-HTOT.n	Total enthalpy for cavity n. (units = J)
CAV-DHR.n	Decay heat rate for cavity n. (units = W)
CAV-M-lay.n	Mass of layer lay in the cavity n. (units = kg)
CAV-T-lay.n	Layer lay temperature in the cavity n. (units = K)
CAV-RHO-lay.n	Layer lay density in the cavity n. (units = kg/m^3)
CAV-THICK-lay.n	Layer lay thickness in the cavity n. (units = m)
CAV-VOL-lay.n	Volume occupied by layer lay in the cavity n. (units = m^3)
CAV-VF-lay.n	Layer lay void fraction in the cavity n. (units = dimensionless)
CAV-MAXRAD.n	Maximum cavity n radius. (units = m)
CAV-MINALT.n	Minimum cavity n altitude. (units = m)
CAV-TMEX.n	Total mass of gas released from cavity n.

	(units = kg)
CAV-MEX-gas.n	Total mass of requested gas released from cavity n. (units = kg)
CAV-QREA.n	Heating rate by chemical reactions in cavity n. (units = W)
CAV-QCNCT.n	Heat loss to concrete in cavity n. Under the assumptions of CORCON-Mod2, this heat goes to the ablation of concrete. (units = W)
CAV-QSURF.n	Heat loss from the surface of debris in cavity n. It may go to either a pool or the atmosphere in the bounding control volume. (units = W)
CAV-R.n.i	Radial coordinate of the i-th body point in cavity n. Available as a plot variable if, and only if, the SHAPEPLOT flag has been set on the CAV_U record (see the description of this flag in Section 2.1). Note the order of the indices. (units = m)
CAV-Z.n.i	Axial coordinate of the i-th body point in cavity n. Available as a plot variable if, and only if, the SHAPEPLOT flag has been set on the CAV_U record (see the description of this flag in Section 2.1). Note the order of the indices. (units = m)
CAV-ASURF.n	Area of upper surface of the debris pool in the cavity n. (units = m^2)
CAV-CRUSTB-lay.n	Thickness of frozen crust on the lower surface. (units = m)
CAV-CRUSTT-lay.n	Thickness of frozen crust on the upper surface. (units = m)
CAV-TSURF.n	Temperature of the upper surface of the debris pool. (units = K)

CAV-MASSERR.n	Mass error in pool. (units = kg)
CAV-ENERGYERR.n	Energy error in pool. (units = J)
CAV-CPUT	Total CPU time used in the CAV package, including computation and I/O. (units = s)
CAV-CPUC	CPU time used for computation in the CAV package. (units = s)
CAV-VIS-lay.n	Layer lay viscosity calculated in the cavity n. (units = kg/m-s)
CAV-RADT.n	Time-dependent melt debris radius for the cavity n. (units = m)
CAV-TLTR.lay.n	R-ave liquid temperature for layer lay in the cavity n. (units = K)
CAV-TLTZ.lay.n	Z-ave liquid temperature for layer lay in the cavity n. (units = K)
CAV-CRUSTR.lay.n	Thickness of frozen crust on the front (R). (units = m)
CAV-EJEC-MAT	Cumulative mass of material MAT transferred to the CAV package. (units = Kg)

5 Control Function Arguments

The CAV package variables that may be used for control function arguments are listed and described below. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section. In the following list, CAVName refers to the cavity name, lay refers to the

desired layer, with HOX referring to the heavy oxide layer, LOX to the light oxide layer, MET to the metal layer, HMX to the heavy mixture layer, and LMX to the light mixture layer. Here heavy and light are in comparison to the metal. Finally gas refers to one of the four gases (H_2, H_2O, CO, or CO_2), and SPName to one of the condensed-phase species (oxides and metals) in the CORCON species list.

CAV-ACTIVe(CAVName)	Activity flag for cavity CAVName. (logical)
CAV-MTOT(CAVName)	Total mass in cavity CAVName. (units = kg)
CAV-HTOT(CAVName)	Total enthalpy for cavity CAVName. (units = J)
CAV-DHR(CAVName)	Decay heat rate for cavity CAVName. (units = W)
CAV-MASS(CAVName,SPName)	Total mass of the requested condensed-phase species in cavity CAVName (summed over all debris layers). (units = kg)
CAV-M(CAVName,Lay)	Mass of the layer. (units = kg)
CAV-T(CAVName,Lay)	Layer temperature. (units = K)
CAV-RHO(CAVName,Lay)	Layer density. (units = kg/m^3)
CAV-THICK(CAVName,Lay)	Layer thickness. (units = m)
CAV-VOL(CAVName,Lay)	Volume occupied by the layer. (units = m^3)
CAV-VF(CAVName,Lay)	Layer void fraction. (units = dimensionless)
CAV-MAXRAD(CAVName)	Maximum cavity radius. (units = m)
CAV-MINALT(CAVName)	Minimum cavity altitude.

	(units = m)
CAV-TMEX(CAVName)	Total mass of gas released from cavity CAVName. (units = kg)
CAV-MEX(CAVName,Gas)	Total mass of the requested gas released from cavity CAVName. (units = kg)
CAV-QREA(CAVName)	Heating rate by chemical reactions in cavity CAVName. (units = W)
CAV-QCNCT(CAVName)	Heat loss to concrete in cavity CAVName. Under the assumptions of CORCON-Mod2, this heat goes to the ablation of concrete. (units = W)
CAV-QSURF(CAVName)	Heat loss from the surface of the debris in cavity CAVName. It may go to either a pool or the atmosphere in the bounding control volume. (units = W)
CAV-TGASMOL(CAVName)	Total moles of gas released from cavity CAVName. (units = mol)
CAV-R(CAVName,i)	Radial coordinate of the i-th body point in cavity CAVName. Available as a plot variable if, and only if, the SHAPEPLOT flag has been set on the CAV_Urecord (see the description of this flag in Section 2.1). Note the order of the indices. (units = m)
CAV-Z(CAVName,i)	Axial coordinate of the i-th body point in cavity CAVName. Available as a plot variable if, and only if, the SHAPEPLOT flag has been set on the CAV_U record (see the description of this flag in Section 2.1). Note the order of the indices. (units = m)
CAV-ASURF(CAVName)	Area of upper surface of the debris pool.

	(units = m^2)
CAV-CRUSTB(CAVName,Lay)	Thickness of frozen crust on the lower surface. (units = m)
CAV-CRUSTT(CAVName,Lay)	Thickness of frozen crust on the upper surface. (units = m)
CAV-TSURF(CAVName)	Temperature of upper surface of the debris pool. (units = K)
CAV-MASSERR(CAVName)	Mass error in pool. (units = kg)
CAV-ENERGYERR(CAVName)	Energy error in pool. (units = J)
CAV-CPUT	Total CPU time used in the CAV package, including computation and I/O. (units = s)
CAV-CPUC	CPU time used for computation in the CAV package. (units = s)
CAV-VISLAY(CAVName,Lay)	Viscosity of layer. (units = kg/m-s)
CAV-RADT(CAVName)	Time-dependent melt debris radius for the cavity n. (units = m)

6 Example Input

Two examples are included in this section. The first example shows typical input for a calculation that begins with an empty cavity and then receives mass from the core package during the calculation, while the second example shows typical input for a calculation that is initiated with mass in the cavity.

6.1 Example for Normal Cavity Input

It is empty at the start of the calculation but receives mass after the core melts through the vessel lower head. The molten core is transferred from the COR package to the

CAV package using the TP package; CAV receives the debris through *out* transfer process name TP102. CORCON type 2 concrete (limestone/common sand) is specified in this example, and 13.5% reinforcing steel is included. The initial cavity is a flat-bottomed cylinder with a radius of 3.0 m and a depth of 5.0 m. The cavity is defined with 95 points, of which 10 are initially on the bottom and 10 are on the 0.1 m radius transition between the bottom and the side. The emissivity of the metal layer is changed from the default value to 0.7.

```
!***** CAVITY INPUT *****
! ***Block: CAV data ****
CAV_INPUT
CAV_ID 'CAVITY 1'
CAV_CV  CV201
!** Concrete type **
CAV_C0  NONSTANDARD    "CORCOR L/CS+FE"
CAV_C1  14 ! NUMBER MATNAME     MASS FRACTION
               1     SIO2           35.8
               2     TIO2           0.18
               3     MNO            0.03
               4     MGO            0.48
               5     CAO            31.30
               6     NA2O           0.082
               7     K2O            1.22
               8     FE2O3          1.44
               9     AL2O3          3.60
              10     CR2O3          0.014
              11     FE             0.135
              12     CO2            21.154
              13     H2OEVAP        2.00
              14     H2OCHEM        2.70
CAV_C2  2340. 1420. 1670. 1500. 298. 0.6
!** Cavity geometry **
!      NRAYS   ZO
CAV_G1  95     0.5
!       ZT   RAD  HIT  RADC
CAV_G2  0.0  3.0  5.0  0.10
!                RW    HBB  NBOT NCORN
CAV_G3   VALUE   4.0   2.0   10   10
! ***Transfer process to introduce debris**
CAV_TP   TP102              TP602
! ***Miscellaneous parameters***
CAV_U  1  ! Number Keyword        Value
               1  EMISS.MET        0.7
***** END OF CAVITY INPUT *****
Input to the TP package will be required in the form
! ***Block: TP data***
TP_INPUT
TP_IN   2 ! NAMETPIN ******* NAMEPAC DIRECT IP2EDF ********
        1  TP101                  COR
        2  TP601                 RNCOR
TP_OUT  2 ! NAMETPOT ***** NPOTOI ****** OUTMTX ******* NMSOT**
        1  TP102           TP101          UIN103          5
        2  TP602           TP601          DEF1           16
!**** NAMEMTX NROW NCOL
TP_MTX UIN103  5  6  ! NROW   Matrix data as a table (NROW,NCOL)
                        1      1.0  0.0  0.0  0.0  0.0  0.0
                        2      0.0  1.0  0.0  0.0  0.0  0.0
                        3      0.0  0.0  1.0  0.0  0.0  0.0
                        4      0.0  0.0  0.0  1.0  0.0  0.0
                        5      0.0  0.0  0.0  0.0  1.0  0.0
```

where it is assumed that the COR package uses *in* transfer process 101 and that RN is active with the default 16 classes. For further discussion, see the TP, RN, and COR users' guides.

By default, the debris will form a single mixed layer. Complete stratification (the CORCONMod2 model) may be enforced by including the record

```
CAV_U  1  ! Number Keyword    Value
               1    MIXING    SUP ! suppress mixing of phases
```

Alternatively, a mechanistic calculation of mixing and reseparation may be specified by including the record

```
CAV_U  1  ! Number Keyword    Value
               1    MIXING    CALC ! perform mechanistic mixing
                                   ! calculation
```

When and if the debris has eroded through the two meters of concrete below cavity name CAVITY 1, the MELCOR calculation will be terminated. It may be restarted by including the record

```
CAV_RA CONTINUE ! ignore axial rupture
```

and will then continue as if there were infinite concrete available. If the user desires that the debris fall into a second cavity and the calculation continue, the record

```
CAV_RA CAVITY CAV7 ! axial rupture with overflow to cavity CAV7
```

should be included in MELGEN input, along with complete input for cavity name CAV7. (MELGEN must run because a cavity cannot be added on restart.)

6.2 Stand-Alone Cavity Example

This example shows typical input for a stand-alone cavity calculation. As in the previous example, cavity name CAVITY 1 is associated with control volume CV201, but in this example, the calculation begins with mass present in the cavity. Two layers are initially present: an oxide layer and a metal layer. Both layers are initially at 2500 K. Basalt concrete is specified, but the composition is modified slightly from the default composition, the initial temperature is raised to 350 K, and the ablation temperature is changed to 1773 K (1500°C). The initial cavity geometry is as in the preceding case.

```
!***** CAVITY INPUT *****
! ***Block: CAV data ****
CAV_INPUT
CAV_ID 'CAVITY 1'
CAV_CV  CV201
!** Concrete type **
CAV_C0  NONSTANDARD    "USER BASALT"
CAV_C1  14 ! NUMBER MATNAME     MASS FRACTION
               1     SIO2           0.7   ! SiO2 content
               2     AL2O3          20.0
               3     CACO3          3.0
               4     CA-OH-2        8.0
               5     H2OEVAP        0.06  ! Water content
```

```
!      DENSCT TSOLCT TLIQCT TABLCT TINCT EMISCT
CAV_C2  2400. 1350. 1650. 1773. 350. 0.6
!** Cavity geometry **
!      NRAYS   ZO
CAV_G1  95     0.5
!       ZT   RAD  HIT  RADC
CAV_G2  0.0  3.0  5.0  0.10
!               RW    HBB  NBOT NCORN
CAV_G3   VALUE   4.0   2.0   10   10
!** Initial layer contents **
CAV_L1   DEFINE  2500. 3 ! NUMBER MATNAM   MASS
                             1     UO2    100000.
                             2     ZRO2   14000.
                             3     FEO    6000.
CAV_L2   DEFINE  2500. 4 ! NUMBER MATNAM  MASS
                             1       ZR    10000.
                             2       FE    70000.
                             3       CR    10000.
                             4       NI    6000.
!***** END OF CAVITY INPUT *****
```

Input must be provided for control volume CV201, from which temperature and pressure boundary conditions for the cavity will be obtained. These properties may be directly specified as tabular functions of time by treating this volume as a time-specified volume. See the CVH Users' Guide for details.

By default, the initial debris will be placed in a single, mixed-phase layer, despite the definition of separate metallic and oxidic layers. The treatment of mixing may be modified by input for the MIXING keyword on a CAV_U tabular record, as described in the previous example.

7 Discussion of Output

The CAV package output should be self-explanatory. Basically, each edit identifies the cavity, its associated control volume, and the status of the cavity (*sleeping* if there is no material presently in cavity, *active* if material is currently present). This is followed by a summary of the major CORCON options in effect and, if the cavity is active, by detailed output (in CORCON format) in several sections as described in the CORCON Reference Manual. This output includes:

(1) A general summary giving the pool configuration, maximum depth and radius, an approximate energy budget, and checks on conservation of mass and energy. As noted in CORCON documentation, the energy budget is approximate and should not be expected to balance exactly. The entry *for changes in pool enthalpy* contains latent heats, and may be significant at times when the pool temperature is nearly constant, but the freezing point of a layer is changing as a result of changing composition. MELCOR calculations frequently require a far larger number of time steps than do calculations with stand-alone CORCON. Therefore, the errors in mass and energy conservation may become larger than a user is accustomed to seeing. Simple accumulation of roundoff would lead to relative errors of approximately SQRT(n) times unit

roundoff after n time steps. This would be roughly 10^{-11} on a typical 64-bit machine or 10^{-4} on a typical 32-bit machine after 10^{4} steps.

(2) Gas generation results, giving generation rates and cumulative releases, both as masses and in moles. Because generation of trace gases has been suppressed, only the major species H_2O, H_2, CO_2, and CO are included.

(3) Geometry of the cavity. Information includes the locations of body points defining the cavity, inclinations of the walls, locations of layer interfaces, and local void fractions.

(4) Details of heat transfer at the interface between the debris pool and the concrete. This includes ablation rates, thickness of the gas film, and temperature of the pool/film interface.

(5) Results of chemical reactions during the time step.

(6) Pool composition.

(7) Layer properties, including thermochemical and transport properties of each occupied layer. Gas-bubble and layer-oriented heat transfer results including crust-model quantities are also given, as are important terms in the energy equation for each layer.

(8) Properties of the individual phases in mixed-phase (HMX or LMX) layers, if present, including mass, density, and melting range.

If the cavity is inactive, the first edit for each calculation will still include geometric and other relevant data. In any case, it will be followed by a summary of concrete properties, including the composition, the composition of reinforcing bar (if present), the melt range, the ablation temperature, and the heat of ablation.

8 Diagnostic and Error Messages

Diagnostic and error messages may be printed either in MELGEN or in MELCOR.

Messages resulting from errors or unrecognized records in user input to MELGEN include the record identifier and sufficient information to allow the user to identify the error. Error messages may be printed during the subsequent calculation of rates in MELGEN or during execution of MELCOR to warn of possibly inaccurate results or indicate the cause of problem termination.

The CORCON manual lists a number of nonfatal errors involving such things as failure of the crust model to converge within the requested tolerance. Because an alternate treatment is provided, and conservation laws are not violated, the default treatment in MELCOR is not to print these messages except to the extended diagnostic file, which is primarily of interest to code developers. In any case, there is little or nothing that a code

user can (or should) do about them. See the EXEC_WARNINGLEVEL input record in the EXEC Package Users' Guide; the messages referred to above have been given a severity level of 3.

There are also several errors beyond which the code cannot continue, and the calculation will be stopped after a restart dump is written. These errors include the appearance of a singular matrix in the chemical equilibrium routine CAV_CCMLTR or in the heat transfer/energy conservation routines CAV_CCENR1, CAV_CCENR2, CAV_CCHLAY, or CAV_CCINTP. The first of these has been reported, the others have not. The user's only option is to restart the calculation with some modification to input—a change in time step is frequently sufficient—to try to avoid the error. It may be necessary to pick the calculation up from an earlier restart rather than the final one.

Another set of fatal errors includes failure in the iterative determination of layer temperatures in subroutine CAV_CCTMPF, which is called from subroutines CAV_CCENR1 and CAV_CCPLAY, or of boundary layer heat transfer in subroutine CAV_CCFILM. Here, also, the user's only option is usually to restart the calculation with some input modification. In the former case (involving subroutine CAV_CCTMPF), the printed output should first be checked to assure that the debris temperatures are reasonable. If they are not, the problem may have been caused by an error in definition of the heat source in CAV, and the associated input and output should be checked carefully.

In the interest of completeness, we note that there are a number of additional points in the CAV package coding where execution will be terminated if an apparent corruption of the database is detected. Examples include, but are not limited to, the overwriting of table bounds in expansion of specific heat tables in subroutine CAV_CCCFND (because the array dimensions are adequate), or the appearance of mass in a mixed-phase layer (because there is no provision for creating such a layer in CAV). Several nonfatal messages not mentioned in this section, such as a warning that the emissivity tables in subroutine CAV_CCEMIS are being extrapolated (because the CAV package does not allow the table option in CORCON) can only result from corruption of the database. Unfortunately, it was determined that these terminations would not be executed or the warnings issued (short of a machine error) in distributed versions of MELCOR. Therefore, if one should be encountered (and is reproducible), the user must first eliminate the possibility that it resulted from an error in a local modification to the code. If it can be demonstrated that this is not the case, the only recourse is to contact the code development team.

Appendix 1

Melcor 2.0	PTFRead	Comments
CAV-MTOT(CAVName) CAVName – the cavity name.	CAV-MTOT.n n – the cavity index.	Total mass in cavity CAVName (n).
CAV-HTOT(CAVName) CAVName – the cavity name.	CAV-HTOT.n n – the cavity index.	Total enthalpy for cavity CAVName (n).
CAV-DHR(CAVName) CAVName – the cavity name.	CAV-DHR.n n – the cavity index.	Decay heat rate for cavity CAVName (n).
CAV-M(CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-M-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Mass of layer Lay in the cavity CAVName (n).
CAV-T(CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-T-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Layer Lay temperature in the cavity CAVName (n).
CAV-RHO(CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-RHO-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Layer Lay density in the cavity CAVName (n).
CAV-THICK (CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-THICK-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Layer Lay thickness in the cavity CAVName (n).
CAV-VOL(CAVName,Lay)	CAV-VOL-Lay.n n – the cavity index;	Volume occupied by layer Lay in the cavity CAVName (n).

Melcor 2.0	PTFRead	Comments
CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	
CAV-VF(CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-VF-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Layer Lay void fraction in the cavity CAVName (n).
CAV-MAXRAD(CAVName) CAVName – the cavity name.	CAV-MAXRAD.n n – the cavity index.	Maximum cavity CAVName (n) radius.
CAV-MINALT(CAVName) CAVName – the cavity name.	CAV-MINALT.n n – the cavity index.	Minimum cavity CAVName (n) altitude.
CAV-TMEX(CAVName) CAVName – the cavity name.	CAV-TMEX.n n – the cavity index.	Total mass of gas released from cavity CAVName (n).
CAV-MEX(CAVName,Gas) CAVName – the cavity name; Gas - one of the four gases (H2, H2O, CO, or CO2).	CAV-MEX-Gas.n n – the cavity index; Gas - one of the four gases (H2, H2O, CO, or CO2).	Total mass of requested gas Gas released from cavity CAVName (n).
CAV-QREA(CAVName) CAVName – the cavity name.	CAV-QREA.n n – the cavity index.	Heating rate by chemical reactions in cavity CAVName (n).
CAV-QCNCT(CAVName) CAVName – the cavity name.	CAV-QCNCT.n n – the cavity index.	Heat loss to concrete in cavity CAVName (n). Under the assumptions of CORCON-Mod2, this heat goes to the ablation of concrete.
CAV-QSURF(CAVName) CAVName – the cavity name.	CAV-QSURF.n n – the cavity index.	Heat loss from the surface of debris in cavity CAVName (n). It may go to either a pool or the atmosphere in the bounding control volume.
CAV-R(CAVName,i) CAVName – the cavity name; i – the body point in the cavity.	CAV-R.n.i n – the cavity index; i – the body point in the cavity.	Radial coordinate of i-th body point in cavity CAVName (n). Available as a plot variable if and only if the SHAPEPLOT flag has been set on the CAV_U record (see the description of this flag in Section **Error! Reference source not found.**). Note the

Melcor 2.0	PTFRead	Comments
		order of the indices.
CAV-Z(CAVName,i) CAVName – the cavity name; i – the body point in the cavity.	CAV-Z.n.i n – the cavity index; i – the body point in the cavity.	Axial coordinate of i-th body point in cavity CAVName (n). Available as a plot variable if and only if the SHAPEPLOT flag has been set on the CAV_U record (see the description of this flag in Section **Error! Reference source not found.**). Note the order of the indices.
CAV-ASURF(CAVName) CAVName – the cavity name.	CAV-ASURF.n n – the cavity index.	Area of upper surface of debris pool in the cavity CAVName (n).
CAV-CRUSTB (CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-CRUSTB-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Thickness of frozen crust on lower surface of the cavity CAVName (n).
CAV-CRUSTT (CAVName,Lay) CAVName – the cavity name; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	CAV-CRUSTT-Lay.n n – the cavity index; Lay – the cavity layer (HOX - the heavy oxide layer, LOX - the light oxide layer, MET - the metal layer, HMX - the heavy mixture layer, and LMX - the light mixture layer).	Thickness of frozen crust on upper surface of the cavity CAVName (n).
CAV-TSURF(CAVName) CAVName – the cavity name.	CAV-TSURF.n n – the cavity index.	Temperature of upper surface of debris pool in the cavity CAVName (n).
CAV-MASSERR(CAVName) CAVName – the cavity name.	CAV-MASSERR.n n – the cavity index.	Mass error in debris pool in the cavity CAVName (n).
CAV-ENERGYERR (CAVName) CAVName – the cavity name.	CAV-ENERGYERR.n n – the cavity index.	Energy error in debris pool in the cavity CAVName (n).
CAV-MTOT(CAVName) CAVName – the cavity name.	CAV-MTOT.n n – the cavity index.	Total mass in cavity CAVName (n).
CAV-HTOT(CAVName) CAVName – the cavity name.	CAV-HTOT.n n – the cavity index.	Total enthalpy for cavity CAVName (n).

Control Function (CF) Package
Users' Guide

The MELCOR Control Function (CF) package allows the user to define functions of variables in the MELCOR database and makes the values of these functions available to other packages in MELCOR. As a result, the user can utilize the time-dependent values of variables that will be calculated during the execution of MELCOR. The CF package has a number of uses, some of which may not be immediately obvious. For example, the pressures in appropriate control volumes may be used to control the opening of a valve or the failure of a blow-out panel; the temperature in a control volume may define the enthalpy associated with a mass source/sink; or the particle loading on a filter may modify the flow resistance in the corresponding flow path. Complicated control logic may be simulated, involving the values of a number of variables in the system. New variables, not ordinarily part of MELCOR output, may be defined for printed or plotted output. The occurrence of specified conditions may be tested and used to initiate generation of additional output (a single line message, a complete list edit, or an additional plot or restart dump) from MELCOR.

This document includes an introduction to control functions, the concepts involved, and how they may be defined and used. It contains a list of the available types of control functions, with the necessary descriptions of each, and describes the relevant MELGEN and MELCOR input. It also presents a number of detailed examples of control function input. This is intended both to aid the new user in getting started and to illustrate a number of techniques which can simplify the task of writing control function definitions.

Contents

CONTENTS **2**

1 INTRODUCTION **4**

1.1 Concept of Control Functions 4
1.2 Applications of Control Functions 5
1.3 Definition of Control Functions 6

2 LIST OF CONTROL FUNCTIONS **7**

2.1 Real-Valued Control Functions 7
2.1.1 Elementary and FORTRAN Functions 7
2.1.2 Derivatives and Integrals 9
2.1.3 Proportional-Integral-Differential Control Function 10
2.1.4 Lag Control Function 10
2.1.5 Hysteresis Control Function 11
2.1.6 Trip Functions 12
2.1.7 User-Defined Functions 14
2.1.8 READ and L-READ Control Functions 16
2.1.9 Larson-Miller Creep Rupture Control Function 16
2.1.10 Pipe Stress Control Function 17
2.2 Logical-Valued Control Functions 18
2.3 User-Defined Control Functions (UDFs) 19

3 USER INPUT REQUIREMENTS **20**

3.1 MELGEN User Input 20

CF_ID – Control Function Definition Record 20
CF_DELETE – Ignore all records associated with CF_ID 21
CF_FORMULA – Formula Text and Arguments 21
CF_CREEP – User-defined Set of Creep Properties 22
CF_SAI – Scale Factor, Additive Constant, and Initial Value of Real Control Function 23
CF_ULB – Upper and Lower Bounds for Real-Valued Control Functions 24
CF_MSC – Miscellaneous Numbers 24
CF_LIV – Initial Value of Logical Control Function 28
CF_CLS – Logical Control Function Classification 28
CF_MSG – Logical Control Function Switching Message 29
CF_ARG – Control Function Arguments 29

3.2 MELCOR User Input 30

4 SENSITIVITY COEFFICIENTS 32

5 PLOT VARIABLES 32

6 CONTROL FUNCTION ARGUMENTS 32

7 EXAMPLE INPUT 32

7.1 Constants 33
7.2 Pressure and Head Differences 33
7.3 Valve Control 34
7.4 Messages and Other Output 35
7.5 New Output Variables 35
7.6 Voting Logic 37
7.7 Homologous Pump Model 38
7.8 Pipe Stress and Creep 39
7.9 Formula type usage 40

APPENDIX 1: CONTROL FUNCTION AND PLOTFILE VARIABLE NAMES 41

APPENDIX 2: USER-DEFINED CONTROL FUNCTIONS – SAMPLE INPUT DECK 42

APPENDIX 3: USER-DEFINED CONTROL FUNCTIONS - DLL IMPLEMENTATION 45

APPENDIX 3A: FUNBOX.F90 REQUIREMENTS 46

1 Introduction

The following subsections give an introduction to control functions as implemented in MELCOR: what they are, how they are used, and how they are defined by user input. Control functions provide a powerful tool, which is intended to alleviate the need to reprogram MELCOR for special cases, permitting the user to effectively modify coding through normal user input.

1.1 Concept of Control Functions

The MELCOR Control Function (CF) package allows the user to define functions of variables in the MELCOR database and makes the values of these functions available to the various physics packages. The values of the control functions are themselves variables in the database, each calculated from the start-of-timestep values of the variables that define its arguments and are therefore also available for printing and/or plotting as desired. Both real-valued and logical-valued control functions are available.

Note that each control function has a unique value at any time, because all of its arguments (other variables in the MELCOR database) are explicit functions of time. It is therefore itself an implicit function of time. This is in contrast to tabular functions, which are functions in the pure mathematical sense of a rule for accepting arguments and returning function values. In order to obtain a value from the Tabular Function (TF) package, a physics package must specify the argument of the function. (This is the reason that tabular functions cannot be plotted directly, although a control function may be defined as a tabular function of time — or any other available argument—and plotted as desired.)

The CF package serves primarily as a utility. It is extremely powerful, but its application is limited to cases where use has been foreseen and appropriate coding has been provided.

(1) In principle, the CF package has access to the entire MELCOR database for use as arguments. This access requires that specific coding be included in the packages that calculate the various data. Therefore, only a limited number of variables in the database are actually available for use as control function arguments; these are listed in the users' guides for the various packages to which the data belong.

(2) The executive (EXEC) package, or any physics package for that matter, may refer to a control function (meaning the value of that control function) by name and use it in any way desired. Here again, the actual use of control functions is limited to cases where specific coding is included in the package that will use the data; this is indicated in the input descriptions in the users' guide of the respective package. Whenever a control function is specified as part of MELGEN input, it is the user's responsibility to see that the function defined is appropriate. The CF package cannot anticipate how a function value will be used and can only check for consistency of its input. Most packages perform only limited error checking in

MELGEN (primarily for existence of the function) and trap only the most obvious errors in function values in MELCOR.

1.2 Applications of Control Functions

A real-valued control function may be used to define an input quantity for a physics package, allowing the input to be a function of current conditions in the system being modeled. The possibilities include:

(1) Rate of mass or enthalpy addition, or temperature of a mass source or sink (negative source) in the Control Volume Hydrodynamics (CVH) package;

(2) Value of an independent thermodynamic variable in a time-specified volume in the CVH package;

(3) Velocity in a time-dependent flow path in the Flow Path (FL) package;

(4) Fraction of a flow path that is open (to represent valves, breaches, blow-out panels, etc.) in the FL package;

(5) Laminar friction coefficient in a flow path segment in the FL package (used to model the effects of filter loading);

(6) Spray flow rate in the Containment Sprays (SPR) package;

(7) Inlet temperature for the dT/dz model in the Core (COR) package;

(8) Fission power in the COR package; and

(9) Partition of decay heat between metallic and oxidic phases in the Cavity (CAV) package.

Logical- or real-valued control functions may be used to initiate or control the operation of components or models as conditions change in a calculation (and in some cases provide the only way to do so). Examples include:

(1) Opening and closing of valves in the FL package;

(2) Operation of pumps in the FL package;

(3) Actuation of spray sources in the SPR package;

(4) Failure of the lower head in the COR package;

(5) Operation of igniters in the BUR package; and

(6) Operation of the passive autocatalytic recombiner (PAR).

Logical-valued control functions may be used to initiate additional MELCOR output as an additional list, plot, restart dump, or short message. This may allow close examination of interesting parts of a calculation or capture peak pressures or temperatures in the list or plot files. It may also permit restarts from potential branch points in an accident sequence or record the time of occurrence of specified events. Control functions may also be used simply to evaluate desired quantities that would not otherwise appear in the list or plot files. Note, however, that MELCOR 1.8.5 allows values of specified control function arguments to be added directly to the plot file, without defining a control function "EQUAL" to that argument. See the EXEC Users' Guide for details.

In all of these cases, complex functions may be built up as desired. For example, a relief valve may be controlled as a function of the pressure difference between two control volumes, including appropriate static head terms. The different setpoints in a multi-bank relief valve system may also be modeled and a block or bypass valve included if desired.

1.3 Definition of Control Functions

A control function is referred to by a user-assigned mnemonic name. Further input defines what type of function it is; for example, the control function may be called 'TEST 1' and evaluated as a square root; the control function may be called 'REACTOR SCRAM' and evaluated using one of several varieties of trips.

Each control function has one or more arguments, which may be real-valued or logical-valued; the number and type of the arguments depend on the specific function involved. The specification of each real-valued argument includes a user-input multiplier and an optional additive constant to be applied to the element of the database to which it refers. That is, if X is a real-valued element of the database that is listed as an available control function argument, the actual argument used in evaluation of a control function has the form A X + B. A value must be specified for A; B has the default value of 0.0. For example, the control function argument, P = CVH-P(CV102), refers to the pressure in control volume CV102 (assuming that this volume exists); the actual argument to be used by a control function could be specified as a_1 = 2.0 * P – 1.7E6.

A real-valued control function may be modified by an overall multiplier and additive constant after the basic function has been evaluated. Thus, using the basic SIN function, the final value of the control function could be specified as 5.3 * SIN (a_1) + 1.8. Optional upper and/or lower bounds on the value may also be specified.

A logical-valued control function may be forced to "latch" so that its value cannot change more than once. It may also be converted to "one-shot," so that it can be true no more than once. Further, it may be made to "alarm" when it changes state, producing a user-defined message in MELCOR output files.

Optional initial values may be specified for each control function. These values may be required to completely define the value of a function with "history" (dependence on past values) such as an integral or a hysteresis function. They may also be used for initialization purposes by other packages during execution of MELGEN. (Note that in some cases the definition might otherwise be circular. For example, the value of a control function defining

a heat transfer coefficient might depend on a temperature that in turn depended on that control function.)

The definition of a control function may also include the input of one or more miscellaneous numbers such as the setpoint for a trip or the user name of a tabular function. This will become clear when particular cases are examined. (When additional data—such as the values of arguments on the previous timestep—must be saved in the CF database, they are simply added to the list of miscellaneous numbers during input processing. Therefore, the number of miscellaneous numbers shown in edits may be greater than the number input.)

The value of a control function is itself available for use as a control function argument so that complicated functions may be built if desired. The user should be aware that control functions are evaluated in the order they were input. Thus, if one control function refers to another that is input after this one, the value used for the latter will always be out of date by one timestep.

2 List of Control Functions

The control functions currently available in MELCOR are described below. In each case, the listing and description correspond to the basic function, and do not include the possible scaling or bounding of real-valued functions or conversion of logical-valued functions to "latches" or "one-shots", which are discussed in Section 1.3. The formal arguments referred to may include scaling from the actual elements of the MELCOR database.

2.1 Real-Valued Control Functions

It is to be understood that, in general, the value of the N-th control function, with n arguments, a_i, and m miscellaneous numbers, c_j, is

$$FUNCT_N = SC_N \, \mathrm{f}(a_1, a_2, ..., a_n; c_1, c_2, ..., c_m) + AC_N$$

where f is the basic function described below, and SC_N and AC_N are input scaling and offset parameters for the control function. Further, the i-th argument has the general form,

$$a_i = SA_i \; X + AA_i$$

where X is a control function argument available in a MELCOR package, and SA_i and AA_i are scaling and offset parameters input to the CF package.

2.1.1 Elementary and FORTRAN Functions

The first group of real-valued functions are elementary and FORTRAN functions. Their definitions should be clear from the following table, which contains a descriptive name, a short name used as the input type, and a mathematical or FORTRAN definition. In some cases, an additional input parameter, referred to as a miscellaneous number, is required to complete the definition of the function.

In the list that follows, the order is not strictly alphabetical; similar functions have been grouped together. Except as noted (for L-A-IFTE), all arguments are real. Miscellaneous numbers may be real or integer, depending on their usage.

Function Name	Input Type	# arg	Mathematical or FORTRAN Definition	Miscellaneous Numbers
Equals	EQUALS	1	$f = a_1$	none
Absolute Value	ABS	1	$f = \lvert a_1 \rvert$	none
Add	ADD	≥ 2	$f = a_1 + \ldots + a_n$	none
Positive Diff	DIM	2	$f = DIM(a_1, a_2)$[a]	none
Multiply	MULTIPLY	≥ 2	$f = a_1 * \ldots * a_n$	none
Divide	DIVIDE	2[b] 1	$f = a_2/a_1$ $f = 1.0/a_1$	none none
Power-Integer	POWER-I	1	$f = (a_1)^J$	1:J = integer exponent
Power-Real	POWER-R	1	$f = (a_1)^R$	1:R = real exponent
Power-Variable	POWER-V	2	$f = (a_1)^{a_2}$	none
Exponential	EXP	1	$f = \exp(a_1)$	none
Natural Log	LN	1	$f = \ln(a_1)$	none
Base 10 Log	LOG	1	$f = \log_{10}(a_1)$	none
Square Root	SQRT	1	$f = \sqrt{a_1}$	none
Cosine	COS	1	$f = \cos(a_1)$	none
Sine	SIN	1	$f = \sin(a_1)$	none
Tangent	TAN	1	$f = \tan(a_1)$	none
Arccosine	ARCCOS	1	$f = \cos^{-1}(a_1)$	none
Arcsine	ARCSIN	1	$f = \sin^{-1}(a_1)$	none
Arctangent	ARCTAN	1	$f = \tan^{-1}(a_1)$	none
Hyperbolic Cos	COSH	1	$f = \cosh(a_1)$	none
Hyperbolic Sin	SINH	1	$f = \sinh(a_1)$	none
Hyperbolic Tan	TANH	1	$f = \tanh(a_1)$	none

[a] DIM(x,y) ≡ max (x-y, 0.0) is a FORTRAN intrinsic function.
[b] DIVIDE may be invoked with one argument or with two.
[c] SIGN (x,y) $\left[\lvert x \rvert, y \geq 0\right]$ $\left[-\lvert x \rvert, y < 0\right]$ is a FORTRAN intrinsic function.
[d] UNIT-NRM differs from SIGNI only in the value for a_1 =0.
[e] TF_{ID} is the MELCOR tabular function with user defined ID, where ID must be a character string.
[f] The first argument of L-A-IFTE is logical, the second and third are real.

Function Name	Input Type	# arg	Mathematical or FORTRAN Definition	Miscellaneous Numbers
Maximum	MAX	≥ 2	f = max(a_1,…,a_n)	none
Minimum	MIN	≥ 2	f = min(a_1,…,a_n)	none
Type one Sign	SIGN	2	f = SIGN(a_2,a_1)[c]	none
Type two Sign	SIGNI	1	f = SIGN(1.0,a_1)[c]	none

Unit Normalize	UNIT-NRM[d]	1	IF (a_1.EQ.0.0) THEN
			f = 0.0
			ELSE
			f = SIGN(1.0,a_1)[c]
			END IF
				none

Tabular Func	TAB-FUN	1	f = TFID (a_1)[e]	ID = Tab. Fcn name

If Then Else	L-A-IFTE	3[f]	IF (a_1) THEN
			f = a_2
			ELSE
			f = a_3
			END IF
				None

READ	READ	0	NA	none
L-READ	L-READ	0	NA	none

[a] DIM(x,y) ≡ max (x-y, 0.0) is a FORTRAN intrinsic function.
[b] DIVIDE may be invoked with one argument or with two.
[c] SIGN (x,y) $\left[|x|, y \geq 0\right]$ $\left[-|x|, y < 0\right]$ is a FORTRAN intrinsic function.
[d] UNIT-NRM differs from SIGNI only in the value for a_1 =0.
[e] TF_{ID} is the MELCOR tabular function with user defined ID, where ID must be a character string.
[f] The first argument of L-A-IFTE is logical, the second and third are real.

2.1.2 Derivatives and Integrals

Two finite-difference approximations to derivatives and one finite-difference approximation to an integral are available as control functions. Each requires two arguments and no miscellaneous numbers, with the first argument being the dependent variable and the second the independent variable. Note that the independent variable is general, and need not be time.

These functions employ values from the MELCOR database at more than one time level. The past values are preserved by the CF package as part of its own database. In what follows, the time level is indicated by a superscript cycle number n, n-1, or n-2. The CF package is called at the end of a MELCOR advancement to calculate values for use on the next step. Therefore, n refers to the end of the just-completed advancement, n-1 to the start of that advancement, and n-2 to the start of the immediately preceding advancement. Note that because of the dependence on past history, the definition of appropriate initial values may be important in initializing these functions. The three functions are

Derivative-Centered Difference (DER-C)

This function evaluates a derivative using information at three time levels in the form

$$f^n = \frac{1}{2}\left(\frac{a_1^{\ n} - a_1^{\ n-1}}{a_2^{\ n} - a_2^{\ n-1}} + \frac{a_1^{\ n-1} - a_1^{\ n-2}}{a_2^{\ n-1} - a_2^{\ n-2}}\right) \tag{2.1}$$

Derivative-Forward Difference (DER-F)

This function evaluates a derivative using information at only two time levels in the form

$$f^n = \left(\frac{a_1^{\ n} - a_1^{\ n-1}}{a_2^{\ n} - a_2^{\ n-1}}\right) \tag{2.2}$$

Integral (INTEG)

This function evaluates an integral using information at two levels. It does not use the additive constant in converting f to FUNCT; if a value for additive constant is input, it must be 0.0.

The integral is calculated as

$$f^n = f^{n-1} + \frac{1}{2}\left(a_1^n + a_1^{n-1}\right)\left(a_2^n - a_2^{n-1}\right) \tag{2.3}$$

2.1.3 Proportional-Integral-Differential Control Function

The Proportional-Integral-Differential control function short input name is PID and requires one argument. It is used in control systems and is defined by the equation,

$$f^n = R_1\, a_1^n + R_2 \int^{t^n} a_1(t)\, dt + R_3 \left.\frac{da_1}{dt}\right|_{t^n} \tag{2.4}$$

where R_1, R_2, and R_3 are input as miscellaneous real numbers. The PID function is evaluated as being equivalent to the composite function

$$\begin{aligned} \mathrm{PID}(a_1) = {} & R_1\, \mathrm{EQUALS}(a_1) + R_2\, \mathrm{INTEG}(a_1, TIME) \\ & + R_3\, \mathrm{DER-F}(a_1, TIME) \end{aligned} \tag{2.5}$$

using equations equivalent to Equations 2.2 and 2.3.

2.1.4 Lag Control Function

The lag function type (designated by the short name LAG) is a basic control theory function for which a function that is passed as an argument, $a_1(t)$, is transformed through the following integral equation.

$$f(t) = \int^{t} \left(\frac{c_2 \cdot a_1(t) - f(t)}{c_1}\right) dt \tag{2.6}$$

Where c_1 is the lag time (seconds) and c_2 is a scaling factor. In differential form, this integral is advanced using the following transform equation.

$$f^{n+1} = \frac{f^{n}\left(1 - \frac{dt}{2c_1}\right) + c_2\left(a_1^{n} + a_1^{n+1}\right)\frac{dt}{2c_1}}{1 + \frac{dt}{2c_1}} \tag{2.7}$$

Example:

```
CF_ID 'LAG'       5028   LAG
CF_SAI 1.0 0.0  0.0
CF_ARG 1
    1      CF-VALU('cSteps') 1.0  0.0  !function, a1, is CF-VALU('cSteps')

!MSC: The first parameter is the time lag (c1) while the second is a multiplier (c2)
CF_MSC  0.5       1.0
```

2.1.5 Hysteresis Control Function

The Hysteresis control function short input name is HYST and requires one argument. It is used to model the type of hysteresis behavior exhibited by components such as relief valves that are opened as the controlling variable (e.g., differential pressure) is increasing and closed as it is decreasing, with a dead band between. A typical hysteresis function is shown in Figure 2.1.

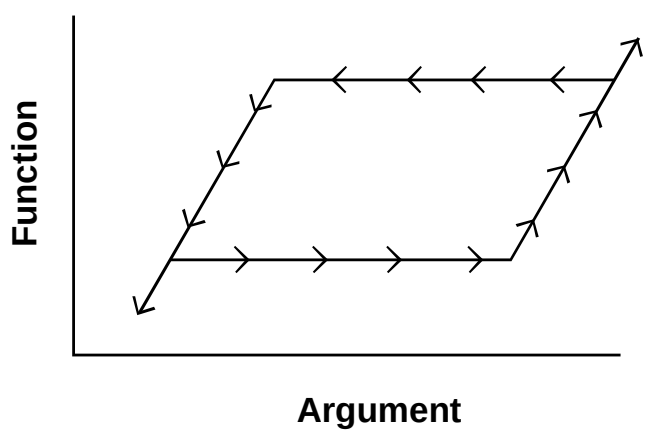


Figure 2.1 Hysteresis Function

The control function value lies between the loading curve, describing the limiting behavior for an increasing argument, and the unloading curve, describing the limiting behavior for a decreasing argument. The value changes only when it is pushed up by the loading curve or pushed down by the unloading curve. Thus, when argument and function value lie within the open loop of Figure 2.1, the function value remains unchanged until the argument has

either increased to encounter the loading curve or decreased to encounter the unloading curve. In either case, the function value will then follow the appropriate bounding curve.

This may be stated somewhat differently as, "when the argument is increasing, the value of the hysteresis function is the maximum of its past value and that defined by the loading curve; when the argument is decreasing, the function takes the minimum of its past value and that given by the unloading curve." That is,

$$f^n = \begin{cases} \max\left[f^{n-1}, \mathrm{f}_{\mathrm{LOAD}}\left(a_1^n\right)\right], & a_1^n \geq a_1^{n-1} \\ \min\left[f^{n-1}, \mathrm{f}_{\mathrm{UNLOAD}}\left(a_1^n\right)\right], & a_1^n < a_1^{n-1} \end{cases} \tag{2.8}$$

The two functions, f_{LOAD} and f_{UNLOAD}, may be specified by tabular functions or other control functions. Two miscellaneous numbers are required. The first identifies the function defining the loading curve, and the second identifies the function defining the unloading curve. The switching flag CF/TF is used to interpret the input name as a control function name or a tabular function name.

Note that if the loading and unloading curves are specified by control functions, it is the user's responsibility to ensure that:

(1) These functions have the same argument as the hysteresis function, and they are set above the hysteresis function (so that they are evaluated first), consistent with Equation 2.8, or

(2) The user understands how the function in Equation 2.8 will be evaluated with the argument of the loading and unloading function not equal to a_1, and that is the desired result.

Necessary historical values are maintained within the CF package database; proper specification of initial function values may be important to achieve the desired results.

2.1.6 Trip Functions

The MELCOR CF package includes various trip functions. Each requires a single argument. In one case, this argument is logical. In all others, the argument is real, and either two or four miscellaneous numbers are required to define setpoints. The latter class of function is used to model switching phenomena involving a dead band (such as a heater which turns on when the temperature falls below T_1 and does not turn off until the temperature has risen above T_2, after which it remains off until the temperature is again below T_1).

In simplest form, such a control has only two states, off and on, and could be represented as a logical function. MELCOR, however, employs a more general implementation where trips have been implemented as real-valued functions. The function value is zero if the trip is off and non-zero if it is on. There are two possible on states: on-forward, indicated by a positive function value, and on-reverse, indicated by a negative value. If a trip is on-forward

or on-reverse, the magnitude of the function value is equal to the time (in seconds) since it was last in some different state and can therefore be used for timing or delay purposes. The terms forward and reverse refer to the direction of counting time when a trip is on; in practice, either type of trip could be used with appropriate (positive or negative) scaling.

The simplest trip, with the short input name of TRIP, is a function of a single logical variable and requires no miscellaneous numbers. It is initially off (with value zero) and remains off until its argument changes state; thereafter, it is on-forward (with positive value) if its argument is true or on-reverse (with negative value) if its argument is false and can never again be off. When the trip is on, the magnitude of the function value is the time elapsed since the last change of state occurred. (For the timestep on which the change took place, this time is taken as roundoff compared to the timestep.)

Of the trips that are functions of a real variable, one of the simplest is TRIP-OFF-FORWARD, abbreviated as T-O-F for input purposes. (The name is intended to indicate that it is off for small argument values and on-forward for large ones.) Two setpoints, S_1 and S_2, $S_1 < S_2$, divide the argument domain into a region where the trip is off, one where it is on-forward, and a dead band in between:

$a_1 \leq S_1$	Trip is off.
$S_1 < a_1 < S_2$	State = previous state, off, or on-forward.
$S_2 \leq a_1$	Trip is on-forward.

The setpoints are input as miscellaneous numbers in the order S_1, S_2. If the trip is off, the function value is 0.0; if it is on-forward, the value is the time elapsed since it was last off. (For the timestep on which the change took place, the time at which the argument crossed the setpoint is estimated by linear interpolation.) The behavior may be compactly summarized as

T-O-F	off	S_1	dead band	S_2	on-fwd

Another simple case involving a real argument is TRIP-REVERSE-OFF, abbreviated as T-R-O. Here, two setpoints, S_1 and S_2, $S_1 < S_2$, divide the argument domain into a region where the trip is on-reverse, one where it is off, and a dead band in between:

$a_1 \leq S_1$	Trip is on-reverse.
$S_1 < a_1 < S_2$	State = previous state, off, or on-forward.
$S_2 \leq a_1$	Trip is off.

If the trip is off, the function value is 0.0; if it is on-reverse, the value is the negative of the time elapsed since it was last off. (Again, the time at which the argument crossed the setpoint is estimated by linear interpolation.) This may be compactly summarized as

T-R-O	on-rev	S_1	dead band	S_2	off

The remaining cases, including several with four setpoints, which define two dead bands, are constructed in exact analogy to the preceding two and should be easily understood from the following summaries. (In all cases, the time at which the argument crosses a setpoint is estimated by linear interpolation.)

T-O-R	off	S_1	dead band	S_2	on-rev				
T-F-O	on-fwd	S_1	dead band	S_2	off				
T-O-F-O	off	S_1	dead band	S_2	on-fwd	S_3	dead-band	S_4	off
T-O-R-O	off	S_1	dead band	S_2	on-rev	S_3	dead-band	S_4	off
T-F-O-F	on-fwd	S_1	dead band	S_2	off	S_3	dead-band	S_4	on-fwd
T-R-O-R	on-rev	S_1	dead band	S_2	off	S_3	dead-band	S_4	on-rev
T-F-O-R	on-fwd	S_1	dead band	S_2	off	S_3	dead-band	S_4	on-rev
T-R-O-F	on-rev	S_1	dead band	S_2	off	S_3	dead-band	S_4	on-fwd

For each of these trips, the setpoints must be input as miscellaneous numbers in the order S_1, S_2, or S_1, S_2, S_3, S_4 as appropriate. Necessary historical data are maintained within the CF database. Because of the dead band, the state of a trip may depend on past history. Therefore, proper definition of initial function values in MELGEN may be important to achieve the desired results.

2.1.7 User-Defined Functions

In order to simplify the addition of complicated functions, a new interface was developed in the F95 version of MELCOR. This interface allows definition of such functions in the formula form on one tabular record rather than constructing them from a series of simpler control functions. The new option may also be used for simple functions; the input is often easier to construct and check.

Such control functions are defined by the new type FORMULA on the CF_ID record. For example,

```
!     CFNAME CFTYPE
CF_ID 'FUNC' FORMULA
```

The function itself is defined by the new tabular record CF_FORMULA for each function of type FORMULA. Two fields are required on the head of the record: number of arguments and the formula text. Then the list of arguments with their values must follow. Record structure is as follows (see Section 3.1 for the record specification):

```
CF_FORMULA  <num of args> <formula text>
            <arg index> <arg name> <arg value>
            <arg index> <arg name> <arg value>
            ...
```

The text of a formula may contain identifier names for arguments, names of elementary or FORTRAN functions, symbols of arithmetical and logical operations, and parentheses ( ) if needed.

The character identifiers of arguments must be composed of one to eight characters. The first must be a letter (A-Z); the remainder may be letters, numerals (0-9), or the underscore character (_).Literal constants (e.g. 2 or 4.7) are not permitted in formula text; constants must be entered as character identifiers and described in the table with the other arguments. There are no reserved words; for example, MAX and AND are permitted argument identifiers. Three function names, UNIT-NRM, L-A-IFTE, and L-L-IFTE, do not conform to the rules for constructing names because they contain hyphens (-) that could be interpreted as subtraction operators. In the interest of backwards compatibility, special treatment is included to allow recognition of these function names.

List of symbols and functions permitted in the formula text include,

Symbols of Arithmetic Operations

```
+
*      /      ^
```

Symbols of Logical Operations

```
<      >      ==     .AND.   .EQV.   .NOT.
<=     >=     \=     .OR.            .NEQV.
```

Elementary or FORTRAN Functions

```
!REAL-VALUED CF****
!of one   argument
ABS                    SQRT           EXP            LN             LOG
SIN                    COS            TAN            ARCSIN         ARCCOS
ARCTAN  SINH           COSH           TANH
SIGNI   UNIT-NRM
!of two   arguments
SIGN           DIM
!of three arguments
L-A-IFTE
!of two or more  arguments
MAX            MIN
!LOGICAL-VALUED CF**
!of three arguments
L-L-IFTE
```

Each row of the table defines an argument from the formula.

```
<arg index> <arg name> <arg value>
```

The argument name is used only in the formula text as an identifier of constant or a control function argument. The value of a control function of type FORMULA (as any other type) may also be used as an argument of some other control function.

The value of FORMULA control function may be, in contrast to control functions of other types, either real or logical. The actual type of such control function is inferred from the formula text.

The input of the other records for a control function of FORMULA type depends on its actual type. An error message will be generated if any errors are found.

The detailed description of record CF_FORMULA can be found in Section 3.1 of this user's guide.

2.1.8 READ and L-READ Control Functions

The READ and L-READ control functions were added to facilitate user interaction with control functions. The READ and L-READ control function are given an initial value that changes only when the user supplies a changedata file that provides an updated value for that CF variable. The changedata file is specified on the EXEC_CFEXFILE record and can provide values for multiple READ control functions. Input required for EXEC_CFEXFILE record is explained in EXEC package user's guide.

The executive will INQUIRE for existence of the changedata file at each timestep when other CFs are being updated, read it if present, and respond accordingly. This file is then deleted once it is processed. The format of records in the changedata file is:

```
CF_ID   CFNAME1  2000.   ! READ CF
CF_ID   CFNAME2_LOGICAL  ON   !L-READ CF
```

A message recording the change in value is issued and, the comment following the (optional) '!' in the record is embedded in the message to allow identification of the change.

The L-READ control function is analogous to the READ control function except it has a logical value indicated by ON (.true.) or OFF (.false.). Initial values are required for both the READ and L-READ control functions. Arguments are not needed and their presence will result in an error message.

An example of input for a READ and L-READ control functions are provided:

```
CF_ID   'CFNAME-READ'  READ
CF_SAI    1.0    0.    0.121  !Initial value is required
CF_ID   'CFNAME-L-READ'  L-READ
CF_LIV    .TRUE.  !Initial value is require
```

2.1.9 Larson-Miller Creep Rupture Control Function

This Larson-Miller Creep Rupture control function short input name is LM-CREEP and requires two arguments. It evaluates the cumulative damage based on the Larson-Miller creep-rupture failure model, which gives the time to rupture, t_R, in seconds as

$$t_R = 10^{\left(\frac{P_{LM}}{T} - C\right)} \tag{2.9}$$

where P_{LM} is the Larson-Miller parameter, T, is the temperature, and C is a property of the material. Note that the correlation is not dimensionless. P_{LM} has dimensions of temperature, and the value of C depends on the units for time. P_{LM} is fit as a function of the effective stress, σ_e, as

$$P_{LM} = \min\left[a_1 \log_{10}(\sigma_e) + b_1, a_2 \log_{10}(\sigma_e) + b_2\right] \tag{2.10}$$

Because stress and temperature are not constant, a fractional lifetime rule is applied, and the damage is calculated as

$$LM\text{-}CREEP(t) = \int \frac{dt}{t_R(t)} \approx \sum \frac{\Delta t_i}{t_R(t_i)} \tag{2.11}$$

Failure is ordinarily assumed to occur when *LM-CREEP*(*t*) = 1.

The material properties to be used are specified by the name included as a miscellaneous number in the control function definition. The recognized values are

(1) A-508, class 2 carbon steel

(2) SS-316, 316 stainless steel

(3) INC-600, Inconel 600

(4) LH-Steel, Default COR LH Steel (SA533B1)

(5) SS-Steel, Default COR SS Steel (SC1604)

Default coefficients for these materials is provided in Table 1. Other materials and parameters may be defined as direct input of five parameters as miscellaneous numbers employing SI units, so that the stress is in Pa=N/m^2; the correlation parameters must be appropriate for temperature in K and time in seconds.

Table 1 Default material properties for LM-CREEP function.

	a_1 [K]	b_1 [K]	C	a_2 [K]	b_2 [K]
A-508	-5335	62291.3	16.44		
SS-316	-7400	81088.4	16.44	-35555.5	322036.2
INC-600	-6296	67129.96	11.44		
SA533B1	-4730	4.81E+04	7.042		
SS-Steel	-7500	8.10E+04	16.44		

2.1.10 Pipe Stress Control Function

The Pipe Stress control function short name is PIPE-STR. It evaluates the maximum stress in a thick-walled cylindrical pipe under internal pressure. This maximum stress occurs at the inner surface, and is given by

$$PIPE\text{ - }STR(t) = \frac{\left(R_o^2 + R_i^2\right)P_i - 2R_o^2 P_o}{\left(R_o + R_i\right)\left(R_o - R_i\right)} \tag{2.12}$$

where P_i is the inner pressure, supplied to the function as the first argument, and P_o is the outer pressure, supplied as the second argument. The inner and outer radii of the pipe (R_i and R_o, respectively) are supplied as miscellaneous numbers.

2.2 Logical-Valued Control Functions

Logical-valued control functions are not amenable to scaling. They may, however, be modified to latch or to function as a one-shot. If a control function is classified as normal, its value will always change in response to change(s) in its argument(s). In contrast, a control function with a latch or one-shot classification becomes uncoupled from its argument(s) after the first change from the initial (MELGEN) value of the function. For the latch, the new value will be retained forever, independent of changes in the function argument(s); for the one-shot, the changed value is retained for one timestep only, after which the function reverts to its initial value on the next timestep and retains that value for the duration of the calculation.

Some logical-valued control function types take logical arguments; others take real arguments. In the latter case, the argument scaling discussed previously should be understood to apply.

In the table below, the order is not strictly alphabetic; similar functions have been grouped together. FORTRAN equivalents are shown in the belief that they will be more familiar than Boolean expressions.

Except as noted (for the arithmetic comparison functions, L-EQ, L-GT, L-GE, and L-NE) all arguments are logical.

Function Name	Input Type	# arg	Mathematical or FORTRAN definition	Miscellaneous Numbers
L-Equals	L-EQUALS	1	f = a_1	none
L-Not	L-NOT	1	f = .NOT. a_1	none
L-Equivalent	L-EQV	2	f = a_1 .EQV. a_2	none
L-Equal	L-EQ	2[a]	f = a_1 .EQ. a_2	none
L-Greater Than	L-GT	2[a]	f = a_1 .GT. a_2	none

[a] Arguments of the arithmetic comparison functions are real.

Function Name	Input Type	# arg	Mathematical or FORTRAN definition	Miscellaneous Numbers
L-Greater or Equal	L-GE	2[a]	f = a_1 .GE. a_2	none
L-Not Equal	L-NE	2[a]	f = a_1 .NE. a_2	none

Function Name	Input Type	# arg	Mathematical or FORTRAN definition	Miscellaneous Numbers
L-AND	L-AND	≥ 2	f = a_1 .AND. … .AND. a_n	none
L-OR	L-OR	≥ 2	f = a_1 .OR. … .OR. a_n	none

L-If Then Else	L-L-IFTE	3	IF (a_1) THEN
			f = a_2
			ELSE
			f = a_3
			END IF
				none

[a] Arguments of the arithmetic comparison functions are real.

2.3 User-Defined Control Functions (UDFs)

Since MELCOR 2.1.6544, the user is now able to create a late-bound library on Windows[1] and define a control function to call user-defined functions in that library. See the appendix for instructions for creating the library and sample source code.

Rules governing user-defined control functions are as follows:

- All user-defined control functions must have five arguments. The user-supplied function implementation may choose to ignore unwanted arguments, but they must still be specified in the function interface.

- There are 10 possible functions. Mnemonics "FUN1"-"FUN10" are used in the CF_ID row in the input file to invoke these functions.

- The late-bound library (`.dll` on Windows, `.so` on Linux) must have one exported function called "FUNN". Each time a UDF is called, MELCOR passes the defined arguments and the UDF name ("FUN1" – "FUN10") identified in input, to the FUNN function. The late-bound library supplied by the user examines this argument at runtime to determine which internal function to call.

- The UDF calculates its result value based on the values of up to five of the inputs passed.

- If UDFs are defined in input, MELCOR requires that a late-bound library called `melcor_user_extension` (`.dll` on Windows, `.so` on Linux) be available either in the MELCOR directory or accessible through normal late-bound extension search rules on the platform used to run MELCOR. These rules can vary by platform, so putting this file in the directory containing your MELCOR and MELGEN executables is recommended.

Linux functionality is planned and will be provided within a few months of this release.

3 User Input Requirements

3.1 MELGEN User Input

The user input for the CF package is described below. One set of records is required for each control function; the actual records required differ for different types of control functions (see Section 2).

CF_ID – Control Function Definition Record

Required

This record defines the control function name and type. The allowed types are described in Section 2.

(1) CFNAME

User-defined control function name

(type = character*16, default = none)

(2) ICFNUM

User defined control function sequence number. Optional. If omitted, a sequence number will be assigned by incrementing the last sequence number by 10.

(type = integer, default = none)

(3) CFTYPE

Control function type (from Section 2) or the character string 'FORMULA'.

(type = character*8, default = none)

Example

```
CF_ID  'MyFunction'    10  EQUALS
```

The data input must be appropriate for the function type.

For real-valued control functions that are not of CFTYPE FORMULA, specify the next records (CF_SAI, CF_ULB, CF_MSC, and CF_ARG) as appropriate.

For logical-valued control functions that are not of CFTYPE FORMULA, specify the next records (CF_LIV, CF_CLS, CF_MSG, and CF_ARG) as appropriate. Note that fields 3 and 4 on record CF_ARG are not input in this case.

If CFTYPE = FORMULA, the additional CF_FORMULA record is required.

If CFTYPE = LM_CREEP, the additional CF_CREEP record may be present.

CF_DELETE – Ignore all records associated with CF_ID

Optional

This record is processed after all records in the input file are read. It will turn off all CF records related to the CFNAME identifier.

(1) CFNAME

User-defined control function name.

(type = character*16, default = none, units = none)

Example

```
CF_DELETE  'MyFunction'
```

CF_FORMULA – Formula Text and Arguments

Required if CFTYPE on CF_ID record equals FORMULA

(1) NCFARG

Number of user-defined arguments (must be >= 1).

(type = integer, default = none)

(2) FTEXT

Formula text.

(type = character*, default = none)

The following data are input as a table with length NUMARG:

(1) NARG

Argument index (is used for sequencing).

(type = integer, default = none)

(2) SHORTNAME

Argument name for usage in formula text.

(type = character*, default = none)

(3) CHARG

Database element identifier (may be real constant, control function, or control function argument).

(type = real or character*, default = none)

SHORTNAME must be composed of one to eight characters. The first must be a letter (A-Z); the remainder may be letters, numerals (0-9), or the underscore character (_). (There are no reserved words; for example, MAX and AND are permitted SHORTNAMEs.)

Example

```
CF_FORMULA  3  (A+B)*C
            1  A  CF-VALU(CF1)
            2  B  CF-VALU(CF2)
            3  C  4.0
```

WARNING: There are two restrictions in the current code version: (1) a logical FORMULA CF that is equal to its single logical argument (which must be the value of some other logical control function) is not permitted, and (2) the single-character 'E' or 'e' is not permitted as a SHORTNAME.

CF_CREEP – User-defined Set of Creep Properties

Optional

This record is used to add creep properties for an additional material to the five materials for which built-in data are available. If this record is present, four fields should be input. The last two fields are optional but they must be input together and will be set to zero by default.

Note that the correlation is not dimensionless. Fit parameters must be consistent with use of SI units (s, Pa, and K, respectively) for the time, t; stress, σe; and temperature, T.

(1) CREEPNAM

Name of user-defined creep property set.

(type = character*32, default = none, units = none)

(2) A1

Inherently negative multiplicative constant, a_1, in fit to P_{LM}, Equation (2).

(type = real, default = none, units = "consistent with SI")

(3) B1

Inherently positive additive constant, b_1,in fit to P_{LM}, Equation (2).

(type = real, default = none, units = "consistent with SI")

(4) C

Constant, C, in fit to t_R, Equation (1).

(type = real, default = 0.0, units = "consistent with SI")

(5) A2

Inherently negative multiplicative constant in fit, a_2 in Equation (2). Optional.

(type = real, default = none, units = "consistent with SI")

(6) B2

Inherently positive additive constant in fit, b_2 in Equation (2). Optional

(type = real, default = 0.0, units = "consistent with SI")

An example of user-defined creep properties for a new material is given below.

Example

```
! PLM = MIN(PLM1,PLM2)
! PLM1 = -7400 log10(stress) + 81088.37
! LOG10(Tremain) = PLM/Temp –16.44370
! PLM2 = -35555.55 log10(stress) + 322036.2
!                      a1     b1      c        a2        b2
CF_CREEP 'User SS316' -7400. 81088.37 16.4437 -35555.55 322036.2
```

CF_SAI – Scale Factor, Additive Constant, and Initial Value of Real Control Function

Required for all types of real-valued control functions except CFTYPE = FORMULA, for which it is optional.

This record defines the multiplicative and additive scaling constants and initial value if needed (the last two fields are optional and may be omitted).

The value of the control function is given by

CFSCAL * Function Value + CFADCN

(1) CFSCAL

Multiplicative scale factor for control function.

(type = real, default = none, units = dimensionless)

(2) CFADCN

Additive constant for control function. Optional field.

(type = real, default = 0.0, units = same as control function)

(3) CFVALR

Real initial value if real-valued control function. Optional field.

(type = real, default = none, units = same as control function)

The initial value of a control function may be user-defined via this record or calculated from the initial database. In cases where the control function involves history (depends on past value), it should be user-defined to assure the desired result. In other cases, identified in the users' guide for the package involved, it should be specified to allow initialization of that package's database.

Note: Field 2 may be omitted if and only if CFVALR is absent. If CFVALR is input, field 2 must be input too.

Example

```
CF_SAI    1.0    0.    0.121
```

CF_ULB – Upper and Lower Bounds for Real-Valued Control Functions

Optional

The user can specify upper and lower bounds for a real-valued control function. If no bounds are input, the function values are limited only by the internal representation of real numbers on the computer used.

(1) ICFLIM

Switch specifying bounds input.

(a) 0 or DEFAULT
No bounds input (default).

(b) 1 or LW
Only lower bound input.

(c) 2 or UP
Only upper bound input (field 2 must be present, but is unused).

(d) 3 or BOTH
Both bounds input.

(type = integer, default = 0, units = dimensionless)

(2) CFLIML

Lower bound, required for ICFLIM = 1, 2, or 3, but unused for ICFLIM = 2.

(type = real, default = none, units = same as control function)

(3) CFLIMU

Upper bound, required for ICFLIM = 2 or 3.

(type = real, default = none, units = same as control function)

Example

```
CF_ULB    BOTH    0.0    1.0
```

CF_MSC – Miscellaneous Numbers

Required for some types of real-valued control functions

Some control functions require input of miscellaneous numbers as noted in Section 2. More than one may be input to a single record if needed.

(1) FIELDS

Appropriate integers, reals, or characters required for the control function, as described in Section 2.

(type = real or integer, default = none, units depend on application)

In case of power-integer function (CFTYPE = POWER-I on record CF_ID), one miscellaneous number of integer type is required.

(1) IEXP

Integer exponent.

(type = integer, default = none)

Example

```
CF_MSC    2
```

In case of power-real function (CFTYPE = POWER-R on record CF_ID), one miscellaneous number of real type is required:

(1) REXP

Real exponent.

(type = real, default = none)

Example

```
CF_MSC    2.3
```

In case of tabular function (CFTYPE = TAB-FUN on record CF_ID), one miscellaneous number of character type is required.

(1) TFNAME

Name of MELCOR tabular function. (All input for this tabular function must be provided. See the TF Users' Guide)

(type = character*16, default = none, units = none)

Example

```
CF_MSC TF84
```

In case of Larson-Miller creep-rupture function (CFTYPE = LM-CREEP on record CF_ID) one miscellaneous number of character type is required.

(1) CREEPNAM

Name of creep property set (default or user-defined).

(type = character*32, default = none, units = none)

Examples

```
CF_MSC 'User SS316' ! user defined
CF_MSC LH-Steel     ! Default COR LH Steel (SA533B1)
```

In case of pipe stress function type (CFTYPE = PIPE-STR on record CF_ID) two miscellaneous numbers are required.

(1) R1

The inner radii of the pipe.

(type = real, default = none, units = m)

(2) R2

The outer radii of the pipe.

(type = real, default = none units = m)

Example

```
CF_MSC 0.37 0.45
```

For trip functions of T-F-O, T-R-O, T-O-F, or T-O-R type (CFTYPE on record CF_ID) two miscellaneous numbers are required; they identify two setpoints with a dead-band in between.

(1) S1

First setpoint.

(type = real, default = none)

(2) S2

Second setpoint.

(type = real, default = none)

Example

```
CF_MSC    -0.1E+10    461945.0
```

For trip functions of T-F-O-R, T-R-O-F, T-O-F-O, T-O-R-O, T-F-O-F, or T-R-O-R type (CFTYPE on record CF_ID), four miscellaneous numbers are required; they identify four setpoints with two dead-bands in between.

(1) S1

First setpoint.

(type = real, default = none)

(2) S2

Second setpoint.

(type = real, default = none)

(3) S3

Third setpoint.

(type = real, default = none)

(4) S4

Fourth setpoint.

(type = real, default = none)

Example

```
CF_MSC    -0.138E+05    -0.1379E+05    0.1079E+05    0.108E+05
```

In case of hysteresis function (CFTYPE = HYST on record CF_ID), four miscellaneous numbers are required; the first and the third identify TF or CF that define the loading and unloading curve correspondingly; the second and the fourth define the function name that defines the loading and unloading curve correspondingly.

(1) CFTF1

Flag indicating function which is used as the first miscellaneous number.

(a) 1 or TF
Tabular function.

(b) -1 or CF
Control function.

(type = integer / character, default = none, units = none)

(2) NCFTF1

Name of tabular function (for CFTF1 = TF).
Name of control function (for CFTF1 = CF).

(type = character*16, default = none, units = none)

(3) CFTF2

Flag indicating function which is used as the second miscellaneous number.

(a) 1 or TF
Tabular function.

(b) -1 or CF
Control function.

(type = integer / character, default = none, units = none)

(4) NCFTF2

Name of tabular function (for CFTF2 = TF).
Name of control function (for CFTF2 = CF).

(type = character*16, default = none, units = none)

Example

```
CF_MSC     TF     TF12     TF     TF13
```

CF_LIV – Initial Value of Logical Control Function

Optional

The initial value of a logical control function may be either user-defined via this record, or calculated from the initial database. In cases where the control function involves history (depends on past value), it should be user-defined to assure the desired result. In other cases, identified in the users' guide for the package involved, it should be specified to allow initialization of that package's database.

(1) LCFVAL

Logical initial value for logical-valued control function.

(a) 0 or FALSE
(b) 1 or TRUE

(type = integer / character*5, default = none, units = dimensionless)

Example

```
CF_LIV FALSE
```

CF_CLS – Logical Control Function Classification

Optional

A logical-valued control function may be converted to a latch or one-shot as described in Section 2.2. A NORMAL control function has a value simply defined by its argument(s). A control function with a LATCH classification will change state only once, retaining its new value thereafter; one with a ONE-SHOT classification will change state for one timestep only, reverting to its initial value thereafter.

(1) CLASS

Classification. May be NORMAL, LATCH, or ONE-SHOT.

(type = character*8, default = NORMAL)

Example

```
CF_CLS NORMAL
```

CF_MSG – Logical Control Function Switching Message

Optional

A logical-valued control function may be made to write a message to the output file, the special message file, and/or the log file whenever its state changes.

(1) MSGFIL

Flag indicating files to which the message is written.

(a) 0 or NO-MESSAGE
Do not write message.

(b) 1 or STANDARD-OUTPUT
Write message to standard output file only.

(c) 2 or FULL-OUTPUT
Write message to standard output file, special message file, and log file but only if timestep is completed successfully.

(type = integer / character*16, default = 0, units = dimensionless)

(2) SWTMSG

Switching message.

(type = character*64, default = none)

Example

```
CF_MSG    FULL-OUTPUT    'drywell failed at personnel airlock'
```

This record must be input for all types except type FORMULA (for which it's prohibited) of control functions:

CF_ARG – Control Function Arguments

Required for all types of control functions except CFTYPE = FORMULA

Each control function requires one or more arguments. Each argument is constructed from an element in the database. The type (REAL or LOGICAL) of each argument is checked in MELGEN, and an error is generated if it does not match that expected for the given control function type (CFTYPE on record CF_ID). Scaling of real arguments is allowed, and the value of argument is given by

ARSCAL * Database Element + ARADCN

The values of ARSCAL and ARADCN must be positioned on the next string after database element identifier. Field 4 is optional and may be omitted.

Scaling of logical arguments is not possible, and the scale factor and constant are not input in this case.

(1) NCFARG

Number of arguments. Must agree with description in Section 2.

(type = integer, default = none)

The following data are input as a table with length NCFARG:

(1) NARG

Argument index is used for sequencing.

(type = integer, default = none, units = none)

(2) CHARG

Database element identifier. Refer to the users' guides for the various packages for permitted values.

(type = character*24, default = none)

(3) ARSCAL

Multiplicative scale factor (only for real-valued functions).

(type = real, default = none, units = dimensionless)

(4) ARADCN

Additive constant (only for real-valued functions).
Optional.

(type = real, default = 0.0, units = same as database element)

Example

```
CF_ARG    2    ! NARG CHARG         ARSCAL   ARADCN
                  1 CF-VALU(CF311)   1.        0.
                  2 EXEC-TIME        0.      461945.0
```

3.2 MELCOR User Input

All MELGEN input records are recognized by MELCOR. New control functions may be added in MELCOR at restart or data associated with an existing control function may be changed when running MELCOR. Record formats are identical for both MELGEN input and MELCOR input.

> Important notice: if user changes some parameters for a previously set CF and doesn't modify optional record for this control function, then the corresponding fields will be taken from the restart file.

Example. Control function name "SRV1" is defined in MELGEN input. For this control function optional CF_ULB record, defining upper and lower bounds for real-valued control functions, is set. In MELCOR input scale factor and additive constant for "SRV1" are redefined. Then all other control function parameters will be taken from MELGEN including upper and lower bound values. This fact is reflected in MELCOR output file showed below together with MELCOR and MELGEN input file pieces.

```
Program MELGEN
CF_ID 'SRV1'      199    T-O-F
CF_SAI 1.    0.0    0.0  ! CFSCAL CFADCN CFVALR (INITIAL VALUE)
       !BOUNDS    LOWER   UPPER
CF_ULB   BOTH      0.0    1.0 ! Min = 0, Max = 1
CF_MSC    0.7+07    0.781514+07       !  Unload/Load Setpoints
CF_ARG    1 ! NARG CHARG        ARSCAL    ARADCN
          1 CF-VALU('P-DIF') 1.0      0.0 ! Wetwell Pressure
END Program MELGEN data
```

```
Program MELCOR
CF_ID 'SRV1'      199    T-O-F
CF_SAI 1.    2.0    3.0  ! CFSCAL CFADCN CFVALR (INITIAL VALUE)
END Program MELCOR data
```

Diagnostic in output file:

```
CONTROL FUNCTION SRV1 IS REAL TYPE T-O-F
  VALUE =  0.0000E+00
  SCALE FACTOR =  2.0000E+00  ADDITIVE CONSTANT =  3.0000E+00
  UPPER BOUND =     1.0000E+00  LOWER BOUND =     0.0000E+00
  NUMBER OF ARGUMENTS =    1
  ARGUMENT NUMBER =    1 HAS CHARACTER IDENTIFIER =CF-VALU('P-DIF')
```

If the user set any record with the optional fields on it in MELCOR and does not provide values for all the optional fields, then MELGEN values will be used for those optional fields that are not input.

The introducing of new control function means that in MELCOR input new CF_ID record appears with new name (not input in MELGEN). Also user may change the type of an existing control function (with name identical to one in MELGEN) but it means that all records for the control function must be input from the beginning (in other words, the changing of CF type is equivalent to the introducing of new CF).

Input deck example

```
! WWWWWWWWWWWW MELCOR input WWWWWWWW
Program MELCOR
!
!* Block: CF data ************************************************************
CF_INPUT ! CF package start record
!*     1 - Number of Control Functions added from the restart
!*     1 - Next CF data *************
!*     1 - Commentary for CF *********
!*
CF_ID 'MyTestCF'    TAB-FUN    ! CFNAME  CFTYPE
```

```
CF_SAI        1.0     0.0     0.0
CF_MSC   'AppliedPower'
CF_ARG   1 !NARG CHARG          ARSCAL         ARADCN
             1 EXEC-TIME         1.0             0.0
END Program MELCOR data
```

4 Sensitivity Coefficients

There are no sensitivity coefficients associated with the CF package.

5 Plot Variables

The plot variables currently included in the CF package are listed below, along with a brief description. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

CFVALU.n	Value of control function index n. At present, only real-valued control functions can be plotted.

Control functions are evaluated in the order they were input. Therefore, if the value of control function n is specified as an argument for control function m, the new value of function n will be used if function m is input after function n, while the old value will be used if function m is placed before function n in the input deck.

6 Control Function Arguments

The control function arguments currently included in the CF package are listed below, along with a brief description. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

CF-VALU(NameCF)	Value of control function NameCF.

Control functions are evaluated in the order they were input. Therefore, if the value of control function n is specified as an argument for control function m, the new value of function n will be used if function m is input after function n, while the old value will be used if function m is placed before function n in the input deck.

7 Example Input

The following examples illustrate input to the CF package and suggest ways in which the package may be used.

7.1 Constants

A constant value may be defined in several ways. Two simple ways to define constant value using scale factors and offsets on the function or on its arguments are:

```
CF_INPUT
CF_ID   'PI'   EQUALS ! Constant value 3.1416
CF_SAI   0.0      3.1416
CF_ARG    1 ! NARG CHARG     ARSCAL    ARADCN
                1 EXEC-TIME    1.0       0.0 !(Argument
                                             ! irrelevant)
or
CF_ID   'PI'   EQUALS ! Constant value 3.1416
CF_SAI   1.0
CF_ARG   1 ! NARG CHARG     ARSCAL    ARADCN
              1 EXEC-TIME     0.0     3.1416 !(Argument
                                             ! irrelevant)
```

In either case, the actual argument specified is irrelevant so long as it is a valid control function argument.

7.2 Pressure and Head Differences

The difference in pressure between two control volumes, CV200 and CV300, perhaps representing primary and containment, can be evaluated as

```
CF_INPUT
CF_ID    'DELTA-P'    ADD ! Pressure difference
CF_SAI    1.0
CF_ARG    2 ! NARG   CHARG    ARSCAL  ARADCN
                1 CVH-P(CV200)   1.0  0.0 !Press in volume CV200
                2 CVH-P(CV300)  -1.0  0.0 !Press in volume CV300
```

Here CVH-P(CVName) is a control function argument in the CVH package, and the pressures used will correspond to the pool surface.

If the static head in the pool in volume CV200 is important, it could be included by modifying CF 'DELTA-P' as follows:

```
CF_ID     POOL-DEPTH    DIM ! Pool depth over junction
CF_SAI    1.0
CF_ARG    2 ! NARG    CHARG        ARSCAL ARADCN
               1 CVH-LIQLEV(CV200)   1.0   0.0  ! Pool surface
                                                ! elevation
               2   EXEC-TIME         0.0   3.57 ! Junction
                                                ! elevation=3.57m
CF_ID    POOL-HEAD    MULTIPLY ! Pressure head in pool
CF_SAI    9.81
CF_ARG    2 ! NARG    CHARG         ARSCAL  ARADCN
               1 CVH-RHO(CV200,POOL)  1.0   0.0 ! Pool density
               2 CF-VALU(POOL-DEPTH)  1.0   0.0 ! Pool depth
                                                ! over junction
CF_ID    'DELTA-P'    ADD ! Pressure difference
CF_SAI    1.0
CF_ARG  3 ! NARG   CHARG       ARSCAL ARADCN
            1 CVH-P(CV200)        1.0 0.0 !Press in volume CV200
            2 CF-VALU(POOL-HEAD)  1.0 0.0 !Press in volume CV300
            3 CVH-P(CV300)       -1.0 0.0 !Press in volume CV300
```

Here CVH-RHO(CVName,POOL) and CVH-LIQLEV(CVName) are the pool density and the pool-surface elevation, respectively, in volume CVName in the CVH package. The junction elevation is 3.57 m (specified as 0.0*EXEC-TIME + 3.57 in CF POOL-DEPTH), and the acceleration of gravity (9.81 m/s^2) appears as the scale factor in CF POOL-HEAD. The relative ordering of the control functions must be as shown so that the value of the pool depth (CF POOL-DEPTH) is current before it is multiplied by the pool density in CF POOL-HEAD, which is done before the head is added to the thermodynamic pressure in CF DELTA-P. Note that it is the position of CF definition in the input deck that is significant.

Clearly, all other head terms, pool, and atmosphere could be included similarly as additional arguments to the ADD function CF DELTA-P.

7.3 Valve Control

The pressure differential given by CF DELTA-P could be used to initiate a primary system failure by opening a valve in an initially closed flow path between volumes CV200 and CV300. For example, the differential could be used to actuate a trip by defining CF RUPTURE input such as

```
CF_ID    RUPTURE    T-O-F ! Trip on pressure diff
CF_SAI    1.0    0.0    0.0 ! Initial value (off)
CF_MSC  -2.0E7  2.0E7 ! Lower (off) and upper (on-fwd) setpoints
CF_ARG    1 ! NARG   CHARG           ARSCAL  ARADCN
               1  CF-VALU(DELTA-P)     1.0     0.0 ! Pressure
                                                   ! difference
```

The use of this trip to actuate the valve must be established in the FL package, i.e., by entering the control function name RUPTURE as variable 7 (NVTRIP) in the table row on record FL_VLV.

The valve state would not change from its initial value (closed) until the pressure differential rose above 20.0 MPa and the control function tripped on-forward. Then, the open fraction would be given by the control function specified by variable 5 (NVFONF) in the table row on the FL_VLV record. (If the differential dropped below the lower setpoint, the open fraction would remain constant with its last calculated value.)

After CF RUPTURE trips on-forward, its value will be the time (in seconds) since the trip. Thus the open fraction could be made a function of time after failure by simply defining the second control function as a function of CF-VALU(RUPTURE).

Alternatively, the pressure differential could be used to control the operation of a relief valve. In this case, no trip would be used (variable 4 in the table row should be set equal to NOTRIP on the FL_VLV record), and the control function name that was input as variable 5 in the table row on the FL_VLV record might define the open fraction of the relief valve area through the hysteresis function

```
CF_ID    VALVE-FRACT    HYST ! Open fract of valve area
CF_SAI    1.0    0.0    0.0  ! Initial value (closed)
CF_MSC    TF     TF100    TF    TF200 ! TF100 for loading and
                                      ! TF200 for unloading
CF_ARG    1 ! NARG    CHARG     ARSCAL ARADCN
```

```
                    1 CF-VALU(DELTA-P) 1.0 0.0 ! Pressure difference
```

The opening characteristic could be made extremely detailed through appropriate definition of the tabular functions and multiple valves included so long as their dead-bands do not overlap. (If the dead bands of a bank of relief valves do overlap, this may be accounted for by summing two or more hysteresis functions in the form of CF VALVE-FRACT, using a below-positioned ADD control function.)

7.4 Messages and Other Output

In either case in Section 7.2, the times at which the valve opened and closed could be recorded in output files using switching messages from logical functions. For example, assuming that the trip option (using CF RUPTURE) was used,

```
CF_ID  VALVE-OPEN     L-GT   ! True if valve open
CF_LIV FALSE    ! Initially false
CF_MSG FULL-OUTPUT 'VALVE OPENED OR CLOSED' ! To out, log,
                                            ! message
CF_ARG    2 ! NARG CHARG             ARSCAL  ARADCN
                1 CF-VALU(RUPTURE)    1.0     0.0 ! Valve trip
                2 EXEC-TIME           0.0     0.0 ! Zero
```

would produce a message every time the valve opened or closed. Inclusion of the record

```
CF_CLS LATCH ! First opening only
```

would terminate messages after the first opening of the valve, in which case a better message might be

```
CF_MSG FULL-OUTPUT 'VALVE OPENED OR CLOSED' ! To out, log,
                                            ! message
Finally, the control function
CF_ID     EDIT-FLAG L-EQUALS ! For record EXEC_EDITCF
CF_LIV    FALSE    ! Initially false
CF_CLS    ONE-SHOT ! True at most once
CF_ARG    1 ! NARG CHARG              ARSCAL  ARADCN
                1 CF-VALU(VALVE-OPEN)   1.0     0.0
```

could be used to trigger a special (complete) edit the first time the valve opened through use of the EXEC_EDITCF record described in the MELCOR/MELGEN EXEC Users' Guide. The one-shot classification prevents more than one such edit from being produced.

7.5 New Output Variables

The values of control functions are included in printed edits and in plot dumps. Thus, the pressure differences, etc., defined in Section 7.2 would appear directly in the printed output of MELCOR and could be plotted as functions of time, if desired. The values printed could be obtained by hand calculations from other edited quantities, but the plots could not be obtained in any other way (using the current version of HISPLT).

Many variables, particularly in the RadioNuclide (RN) package, are available as control function arguments but not as plot variables. This is because the size and complexity of the database make it impractical to include all quantities in the plot file. In MELCOR versions

through 1.8.4, these variables could be plotted only by defining a control function equal to the argument and then plotting the value of that control function. This is no longer necessary, because the values of specified control function arguments can be added directly to the plot file using EXEC_PLOT input records without defining a control function. In addition to simplification of the input, this allows the variables to be plotted using their mnemonic names. See the EXEC Package Users' Guide for details.

There will still be cases where the user will need to define a control function simply to make available the value of a quantity that would otherwise not appear in the output, but only when some arithmetic or logic is involved. As a simple example involving hydrodynamic variables, the peak pressure in a control volume could be evaluated by a single control function as

```
CF_ID   PEAK-PRESS   MAX  ! Peak pressure in CV202
CF_SAI  1.0  0.0  0.0     ! Initialize to 0.0
CF_ARG   2 ! NARG    CHARG           ARSCAL  ARADCN
               1  CVH-P(CV202)         1.0    0.0  ! Pressure in
                                                   ! CV202
               2  CF-VALU(PEAK-PRESS)  1.0    0.0  ! Previous
                                                   ! peak
```

that evaluates the maximum of the pressure in control volume CV202 and the previous value of that maximum.

Another way to do this, and to capture the time at which the peak occurred, is

```
CF_ID    NEW-GT-OLD    L-GT ! True if p .gt. old max
CF_LIV   TRUE  ! Initialize true
CF_ARG    2 ! NARG    CHARG         ARSCAL  ARADCN
                1 CVH-P(CV202)        1.0     0.0 ! Pressure in
                                                  ! CV202
                2 CF-VALU(PEAK-PRESS) 1.0     0.0 ! Previous
                                                  ! peak press
CF_ID    PEAK-PRESS    L-A-IFTE ! Peak pressure in CV202
CF_SAI    1.0
CF_ARG  3 ! NARG CHARG             ARSCAL            ARADCN
             1 CF-VALU(NEW-GT-OLD)  1.0 0.0 ! New .gt. old peak
             2 CVH-P(CV202)         1.0 0.0 ! Pressure in CV202
             3 CF-VALU(PEAK-PRESS)  1.0 0.0 !Previous peak press
CF_ID    T-AT-PEAK    L-A-IFTE ! Time at peak pressure
CF_SAI    1.0
CF_ARG  3 ! NARG CHARG           ARSCAL ARADCN
             1 CF-VALU(NEW-GT-OLD)  1.0 0.0 ! New .gt. old peak
             2 EXEC-TIME            1.0 0.0 ! Time
             3 CF-VALU(T-AT-PEAK)   1.0 0.0 ! Time at previous
                                            ! peak
```

Because CF NEW-GT-OLD is given the initial value TRUE, CF PEAK-PRESS and CF T-AT-PEAK will be properly initialized to the values of their second arguments, CVH-P(CV202) and EXEC-TIME, respectively. The uninitialized third arguments (the previous values of these control functions) will not be used. Otherwise, the user would have to initialize (correctly) both CF PEAK-PRESS and CF T-AT-PEAK, in which case no initial value for CF NEW-GT-OLD would be needed. The values of all three control functions

would be available in printed output. The real-valued functions could also be plotted if desired.

7.6 Voting Logic

Control logic may be simulated in many ways, and there are often available techniques that employ far fewer control functions than a simple translation of the logic. For example, a two-of-four voting control could be modeled using four logical functions for the individual channels, six more to test all combinations of two true signals, an eleventh to "OR" these six possibilities, and (usually) a twelfth to generate a real value from this final logical one.

A more compact, but less obvious, implementation is shown below for the case where the individual channels involve detecting a pressure greater than a threshold.

```
! Test each channel. Value =1 if Pn.ge.thresh, else =0
!
CF_ID    P1.GT.TH    SIGNI ! 1 or 0 form p1 test
CF_SAI    .5   .5
CF_ARG  1 ! NARG CHARG    ARSCAL ARADCN
             1 CVH-P(CV1)   1.0 -1.02E5 ! 1.02E5 is thresh
CF_ID    P2.GT.TH    SIGNI ! 1 or 0 from p2 test
CF_SAI    .5   .5
CF_ARG  1 ! NARG CHARG    ARSCAL ARADCN
             1 CVH-P(CV2)   1.0 -1.02E5 ! 1.02E5 is thresh
CF_ID    P3.GT.TH    SIGNI ! Same for p3
CF_SAI    .5   .5
CF_ARG  1 ! NARG CHARG    ARSCAL ARADCN
             1 CVH-P(CV3)   1.0 -1.02E5 ! 1.02E5 is thresh
CF_ID    P4.GT.TH    SIGNI ! Same for p4
CF_SAI    .5   .5
CF_ARG  1 ! NARG CHARG    ARSCAL ARADCN
             1 CVH-P(CV4)   1.0 -1.02E5 ! 1.02E5 is thresh

!
! Vote 2 of 4. Result is 0.0 or 1.0
!
CF_ID    2-OF-4    ADD ! Value = 2*N – 2.5
CF_SAI    2.    -2.5
CF_ULB    BOTH  0.0   1.0 ! Limit between 0 and 1
CF_ARG 4 ! NARG     CHARG       ARSCAL ARADCN
             1 CF-VALU('P1.GT.TH') 1.0 0.0 !1 or 0 for channel 1
             2 CF-VALU('P2.GT.TH') 1.0 0.0 !1 or 0 for channel 2
             3 CF-VALU('P3.GT.TH') 1.0 0.0 !1 or 0 for channel 3
             4 CF-VALU('P4.GT.TH') 1.0 0.0 !1 or 0 for channel 4
```

Note the usage of multiplicative and additive constants and of limits. For each channel (CF Pn.GT.TH) the function value is –1.0 or +1.0 before scaling and 0.0 or 1.0 after. In the vote conducted by CF 2-OF-4, for the case of

0	1	2	3	4	pressures over threshold,

the function value is

0.0	1.0	2.0	3.0	4.0	before scaling,

-2.5	-0.5	+1.5	+3.5	+5.5	after scaling, and
0.0	0.0	1.0	1.0	1.0	after limits are imposed.

CF 2-OF-4 could now be used to initiate a trip and start a timer. For example,

```
CF_ID    VOTE-TRIP    T-O-F ! Trip on 2-of-4 vote
CF_SAI    1.0   0.0   0.0   ! Initially value (off)
CF_MSC    0.1   0.9   ! Lower (off) and Upper (on) setpoints
CF_ARG    1 ! NARG CHARG              ARSCAL ARADCN
                1 CF-VALU('2-OF-4')     1.0    0.0 ! 2-of-4
```

Note that the scaling and bounding of control functions Pn.GT.TH (n =1…4) and 2-of-4 are not essential if their only use is to actuate the trip; the setpoints of CF VOTE-TRIP could be changed to work correctly with the unscaled, unbounded values. However, the scaling costs little and makes it easier to check the control logic.

7.7 Homologous Pump Model

The QUICK-CF pump model in the FL package uses a control function to calculate the pressure developed by a pump. The following input indicates how a control function can be constructed to yield a homologous pump model similar to that used in TRAC and RELAP.

```
!Homologous pump model for flow path FL2
!CF DELTA-P = delta-p = density*grav*speed**2 * fcn (flow/speed)
!
CF_ID SPEED ! any real CF, (Normalized) pump speed
...
!
CF_ID     FLOW/SPEED DIVIDE   ! (Normalized) flow/speed
CF_SAI    1.0
CF_ARG  2 ! NARG   CHARG     ARSCAL  ARADCN
            1 CF-VALU(SPEED)  1.0 0.0 ! Speed
            2 FL-VEL(FL2,P)  0.05 0.0 ! 1/.05 is v at rated flow
!
CF_ID     PUMP-HEAD   TAB-FUN   ! Fcn (flow/speed) in m
CF_SAI    1.
CF_MSC    PUMP-CURVE
CF_ARG  1 ! NARG CHARG           ARSCAL  ARADCN
         1 CF-VALU('FLOW/SPEED')   1.0     0.0 ! (Normalized)
                                               ! flow/speed

CF_ID DELTA-P MULTIPLY
CF_SAI    9.81          ! 9.81 is grav
CF_ARG    4 ! NARG CHARG       ARSCAL ARADCN
            1 CVH-RHO(CV3,POOL)  1.0   0.0 ! Upstream volume is
                                           ! CV3
            2 CF-VALU(PUMP-HEAD) 1.0   0.0 ! Head in m
            3 CF-VALU(SPEED)     1.0   0.0 ! Speed/nom_speed
            4 CF-VALU(SPEED)     1.0   0.0 ! Speed/nom_speed
...
TF_INPUT
TF_ID PUMP-CURVE    1.0
! Table of n points. Norm flow/speed Head (m)
TF_TAB  n ! NTFPAR    X    Y
              1      a.a  x.x
              2      b.b  y.y
            ...    ...   ...
```

The input shown is appropriate for the normal region of pump operation (positive rotation, forward flow), when the fluid is liquid only. The calculated pressure boost will be far too great for two-phase or vapor flow. Because the control function defines the pressure directly rather than as a head, it must be reduced in these cases by the density ratio (in addition to any two-phase degradation factor). To do so, the density used as the first argument of CF DELTA-P should be generalized to represent the actual flow density multiplied, if desired, by a two-phase degradation term. As a first approximation, it could simply be set to zero if the flow path void fraction—available as control function argument FL-VOID(NameFP)—is non-zero.

It might be necessary to incorporate some time-averaging through integration if oscillations are observed.

In RELAP, the head curve in the normal pump region is input in two parts, using

HEAD = SPEED**2*FCN1(FLOW/SPEED) for FLOW/SPEED < 1 and

HEAD = FLOW **2*FCN2(SPEED/FLOW) for SPEED/FLOW < 1.

(the former is referred to as HAN and the latter as HVN), but both may be accommodated in CF PUMP-HEAD. The curve can be further extended to the energy dissipation region (positive rotation, negative flow), corresponding to HAD and HVD. In this case, the density argument in CF DELTA-P should also reflect the direction of flow.

The input assumes that the rotational speed of the pump is available as a control function value. In general, this will be a tabular function of time, but a truly determined user could calculate it by integration of the angular momentum equation for a pump rotor, using still more control functions and the homologous torque curves. If reversal of the pump is anticipated, it is necessary (at least) to avoid the possibility of division by zero in CF FLOW/SPEED.

This is probably best done by explicitly using the alternative representation of the head curve in those regions where SPEED/FLOW < 1 (HVN, etc.).

7.8 Pipe Stress and Creep

The PIPE-STR CF and LM-CREEP CF can be used to calculate the creep failure of a pipe under pressure-induced stresses. The value of control function Stress is the cumulative damage to the pipe, calculated in the Larson-Miller model; failure is ordinarily assumed when its value exceeds 1.0.

```
CF_ID Stress PIPE-STR
CF_SAI 1.0  0.0
CF_MSC 0.37 0.45
CF_ARG 2 ! NARG CHARG    ARSCAL   ARADCN
           1 CVH-P(CV500)  1.      0. ! Inner pressure (hot leg)
           2 CVH-P(CV8)    1.      0. ! Outer pressure
                                      ! (containment)

CF_ID 'Pipe Creep' LM-CREEP
CF_SAI 1.0  0.0
```

```
CF_MSC SS-316   ! default data of 316 Stainless steel
CF_ARG 2 ! NARG CHARG    ARSCAL   ARADCN
           1 CF-VALU(Stress)    1.  0. ! Stress
           2 HS-TEMP(HS50001,2) 1.  0. ! Temp, node 2
```

7.9 Formula type usage

The following two examples illustrate possible input for the Larson-Miller creep-rupture failure expressions using record CF_FORMULA. The third example shows the FORMULA control function of logical type.

The first input defines Larson-Miller parameter P_{LM} (see Section 2.1.7 for the formula expression). Input that defines the parameter may be as follows:

```
!           CFNAME CFTYPE
CF_ID       SIG_E  ADD       !used in formula below and
...                          !must be defined correctly
CF_ID   P_LM   FORMULA  ! new record to define the formula
!          NUM OF ARGS FORMULA TEXT
CF_FORMULA  5 MIN(A1*LN(SE)+B1,A2*LN(SE)+B2)
! data table for arguments
! N  ARGNAME VALUE
  1  A1   1.e5  ! Constant
  2  B1   2.e-4 ! Constant
  3  A2   0.5   ! Constant
  4  B2 ...     ! Constant
  5  SE    CF-VALU(SIG_E) ! The effective stress
```

The second input defines the time to rupture, t_R. (see Section 2.1.7 for the formula expression). Input that defines the parameter may be as follows:

```
!           CFNAME      CFTYPE
CF_ID       t_R         FORMULA
!           NUM OF ARGS FORMULA TEXT
CF_FORMULA  5           U/(A^(PLM/T-C))
!          N  ARG NAME VALUE
            1  A     10.                ! Constant
            2  C     2.e-5              ! Constant
            3  PLM   CF-VALU(P_LM)      !P_LM is defined above
            4  T     HS-TEMP(HS50001,2) ! Temperature
            5  U     1.0                ! Constant
```

The last example is input of a logical-valued function.

```
!           CFNAME      CFTYPE
CF_ID       P           ...    ! used in formula
...
!           CFNAME      CFTYPE
CF_ID       Break_Cond  FORMULA
!           NUM OF ARGS FORMULA TEXT
CF_FORMULA  5           P > Ro*R*T + Po
!           N  ARG NAME VALUE
            1  R        8314.           !constant
            2  P        CF-VALU(P)      !P is defined above
            3  T        CVH-TVAP(CV200) !Atmosphere temp.
            4  Ro       2.e-5
            5  Po       1.e5
```

Appendix 1: Control Function and Plotfile Variable Names

Melcor 2.x	PTFRead	Comments
CF-VALU(NameCF) NameCF – the control function name.	CFVALU.n n - the control function index.	Value of control function NameCF (n).

Appendix 2: User-Defined Control Functions – Sample Input Deck

MELCOR requires a late-bound library (`.dll` on Windows, `.so` on Linux) to be present to call when the FUN1-FUN10 control functions are defined. The following input block defines a call to a user-defined function called FUN1. It was extracted from the `cf_funn_test.inp` input deck shown further below. You can also request this entire deck from the MELCOR software development team.

```
CF_INPUT

!    Variable user name, ID,      CF Type

CF_ID 'User-coded FUN1'  101      FUN1

!         cfscal cfadcn  cfvalr

CF_SAI    1.0    0.0     !  0.10 ! No Initial Value

CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN

              1    EXEC-TIME    1.0    0.00

              2    EXEC-TIME    1.0    0.00

              3    EXEC-TIME    1.0    0.00

              4    EXEC-TIME    1.0    0.00

              5    EXEC-TIME    1.0    0.00
```

In this example, EXEC-TIME is passed, without modification, as all five arguments. At each timestep, MELCOR calls the exported "FUNN" routine supplied by the user. That routine is expected (in this case) to evaluate its CFTYPE argument, determine that "FUN1" is the requested function, and provide and call a custom implementation of the FUN1 function.

Here is a complete, simple input deck for calling user-defined control functions (UDFs). If you need an ASCII version of this file you can request `_cf_funn_test.inp` from the MELCOR software development team. It contains a very basic input block for each function, FUN1-FUN10:

```
MEG_DIAGFILE 'FUNn_Test.dia'
MEG_OUTPUTFILE 'FUNn_Test.outg'
MEG_RESTARTFILE 'FUNn_Test.rst'
MEL_OUTPUTFILE 'FUNn_Test.out'
MEL_DIAGFILE 'FUNn_Test.dia'
PLOTFILE 'FUNn_Test.PTF'
MEL_RESTARTFILE 'FUNn_Test.rst' NCYCLE     0
Program MELGEN
```

```
EXEC_INPUT
EXEC_TITLE 'FUNn_Test'
CF_INPUT
CF_ID 'User-coded FUN1'  101      FUN1
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0     !  0.10 ! No Initial Value
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00

CF_ID 'User-coded FUN2'  102      FUN2
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN3'  103      FUN3
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0     !  0.10 ! No Initial Value
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00

CF_ID 'User-coded FUN4'  104      FUN4
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN5'  105      FUN5
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN6'  106      FUN6
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
```

```
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN7'  107      FUN7
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN8'  108      FUN8
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN9'  109      FUN9
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0    0.00
              2    EXEC-TIME    1.0    0.00
              3    EXEC-TIME    1.0    0.00
              4    EXEC-TIME    1.0    0.00
              5    EXEC-TIME    1.0    0.00
CF_ID 'User-coded FUN10'  110      FUN10
!         cfscal cfadcn  cfvalr
CF_SAI    1.0    0.0      ! 0.00
CF_ARG    5 ! NARG CHARG        ARSCAL ARADCN
              1    EXEC-TIME    1.0   0.00
              2    EXEC-TIME    1.0   0.00
              3    EXEC-TIME    1.0   0.00
              4    EXEC-TIME    1.0   0.00
              5    EXEC-TIME    1.0   0.00
END Program MELGEN
Program MELCOR
EXEC_INPUT
EXEC_TITLE 'FUNn_Test'
EXEC_TEND  200.0
EXEC_TIME  1
           1  0.   .1  0.01   1.  0.1 1000.
EXEC_CPULEFT  20.0
EXEC_CPULIM   1800.
CF_INPUT
END Program MELCOR
```

Appendix 3: User-Defined Control Functions - DLL Implementation

MELCOR archives beginning with the 2.1.6544 release contain a sample DLL project in the `tools\UDFs` folder that can be modified for your use. Because creating and troubleshooting DLLs and shared objects is an advanced programming topic, it is highly recommended that you employ a qualified programmer with experience in this task. The MELCOR software development team is not funded for technical support of this feature. Besides providing sample code and this documentation, the MELCOR team is unable to provide any additional free technical support.

To prevent issues loading and calling your functions, static binding to library dependencies is recommended. This documentation doesn't cover implementation details for activating static binding. There are multiple compilers and linkers with different ways of ensuring/activating it, and troubleshooting can be problematic even for those with requisite experience. The project included in the distribution archive provides a good starting point and is already set up to create statically-linked libraries.

An Intel Fortran Compiler solution is provided (melcor_user_extension.sln). This solution can be modified by the user to create desired functionality. It includes the following source files:

Source File	Description	Notes
funbox.f90	Main sample DLL file	Contains implementation of FUN1-FUN10 routines and main late-bound entry point: FUNN.
c_string_copying.f90	String conversion routines	Routines for conversion of Fortran strings to C and vice-versa
M_kind.f90	Contains definitions for data types used by MELCOR	Ensures consistency between data types used by MELCOR late-bound library
uccf_externals.f90 UserCodedCF.f90	Not yet implemented fully in MELCOR	UCCF (generalized User Coded Control Functions) are an unscheduled extension to MELCOR. This code should be left unmodified and is included for future feature compatibility.

Appendix 3a: funbox.f90 Requirements

When creating a DLL, these preprocessor directives must be defined: UCCF and FUNN. You must also ensure the Preprocess Source File flag is activated.

It is recommended that you put your implementation into your own source files. This keeps your code from having to change when new versions of funbox.f90 are released. To do this is relatively simple:

- Define FUNN_ROUTINES_ARE_EXTERNAL
- Copy the FUN1 – FUN10 routines from funbox.f90 to a separate source file in your project (not a module) and modify them in that location
- When FUNN is called by MELCOR, calls to your implementation will be handled automatically by the linker.

For simplicity, don't define FUNN_ROUTINES_ARE_EXTERNAL. Then the FUN1 – FUN10 function bodies in funbox.f90 can be directly modified.

Condenser (CND) Package
Users' Guide

The MELCOR Condenser (CND) package models the effects of the Isolation Condenser System (ICS) and the Passive Containment Cooling System (PCCS); both use heat exchangers submerged in large water pools. Several older boiling water reactors (BWRs) and the proposed new simplified boiling water reactor (SBWR) contain isolation condensers to condense steam created in the core and return it to the primary system. Only the SBWR, however, includes the passive containment cooling system to provide steam suppression in the drywell during a loss-of-coolant accident (LOCA) or when depressurization valves equalize the pressures of the reactor vessel and containment. This equalization is required so water can drain to the reactor vessel from the gravity-driven cooling system (GDCS) pools located several meters above the top of the core. The CND package is a subpackage within the ESF package. This users' guide provides basic information needed to run the CND model with the rest of MELCOR, including a detailed explanation of user input for MELGEN and MELCOR. Required and optional input, sensitivity coefficients, control function arguments, plot variables, and error messages are all covered.

More detailed information on the technique for simulating the effects of the operation of the two condenser types can be found in the CND Package Reference Manual.

Contents

LIST OF TABLES **2**

LIST OF FIGURES **3**

1 INTRODUCTION **4**

2 INPUT REQUIREMENTS **6**

2.1 MELGEN Input 6

CND_PCCS01 – CVH Volumes that Interface with the PCCS 6
CND_PCCS02 – Tabular Function for the Variation of Capacity 7
CND_PCCS03 – Geometric Input for the PCCS Vent Line 9
CND_PCCS04 – PCCS Unit(s) Description 9
CND_ICS01 – CVH Volumes that Interface with the ICS 10
CND_ICS02 – Tabular Function for the Variation of ICS Capacity 12
CND_ICS03 – Geometric Input for the ICS Vent Line 13
CND_ICS04 – ICS Unit(s) Description 13

2.2 MELCOR Input 15

3 SPECIAL OUTPUT DATA FILES **15**

4 PLOT VARIABLES **19**

5 CONTROL FUNCTION ARGUMENTS **21**

6 EXAMPLE INPUT **21**

6.1 Example MELGEN Input 21

7 CND MODEL OUTPUT **29**

8 DIAGNOSTIC AND ERROR MESSAGES **29**

List of Tables

Table 3.1 Description of Variables in ICS/PCCS Data Files 17

List of Figures

Figure 1.1 Typical Passive Containment Cooler Condenser (left) and Isolation Condenser (right) 5
Figure 1.2 Example Flow Diagram of the Passive Containment Cooling System 5

1 Introduction

The MELCOR Condenser (CND) package models the effects of the Isolation Condenser System (ICS) and the Passive Containment Cooling System (PCCS); both use heat exchangers submerged in large water pools. The input for these condenser models assume that the MELCOR code should adequately represent the condenser effects under the boundary conditions that accidents would impose. The MELCOR calculation is not intended to predict the performance of these heat exchanger-condenser systems, based on basic physical considerations; more sophisticated thermal hydraulic codes accomplish that task, which is not compatible with the role of a fast-running Probabilistic Risk Assessment (PRA) tool. Furthermore, test calculations performed with MELCOR demonstrate that attempts to use the basic code *building-block* approach—to connect control volumes, flow paths, and heat sink structures as necessary to directly simulate the heat exchanger-condensers—will result in code difficulties; these difficulties include oscillations in the predicted flows and energy exchanges, a demand for extremely small timesteps, and impractical large central processing unit (CPU) and wall clock time consumption.

The condenser model was originally developed to model the PCCS and the ICS. Several older boiling water reactors (BWRs) and the new simplified boiling water reactor (SBWR) designs contain isolation condensers to condense steam created in the core and return it to the primary system. However, only the SBWR utilizes the PCCS to provide steam suppression in the drywell during a loss of coolant accident (LOCA) or when the depressurization valves are used to reduce the differential pressure between the reactor vessel and containment. Therefore, the model has an implied geometry and also contains limitations set by the normal operating characteristics of the PCCS and ICS. Their geometries are shown in Figure 1.1 and Figure 1.2.

For the ICS, the source volume is assumed to be the upper portion of the reactor vessel; and the drain is assumed to be the reactor vessel annulus. For the PCCS, the source volume is assumed to be the volume representing the upper drywell and the drain volume is assumed to be any Control Volume Hydrodynamics (CVH) located in the drywell. In either case, the drain volume is at approximately the same pressure as the source volume; therefore, the gravity head is sufficient to allow free flow of condensate from the condenser. Note that a flow calculation—to flush the condensers of noncondensible gases—is performed based on the differential pressure between the *source* volume and the vent volume. However, back flow through either the drain or vent line is not permitted.

Although the PCCS and ICS models were designed for SBWR applications, allowances have been made in the CND package input processing so that the PCCS and ICS condenser models can be used for either BWR or pressurized-water reactor (PWR) reactor types.

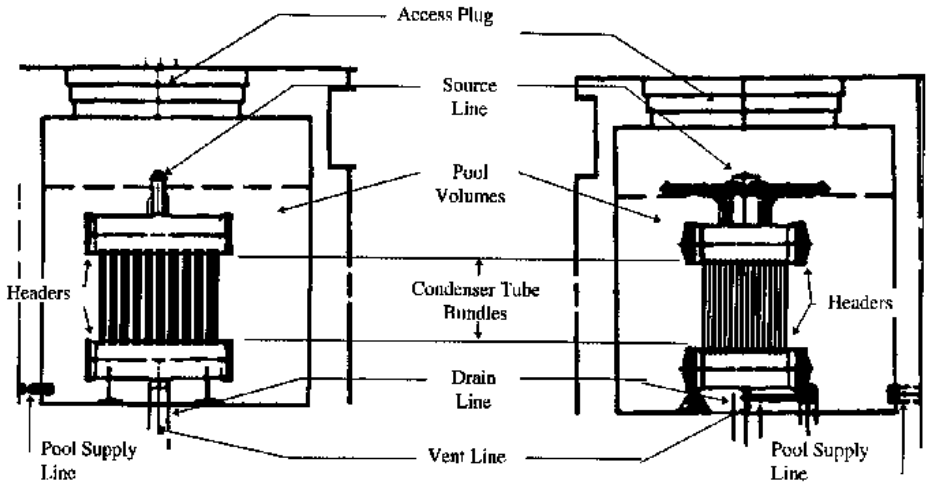


Figure 1.1 Typical Passive Containment Cooler Condenser (left) and Isolation Condenser (right)

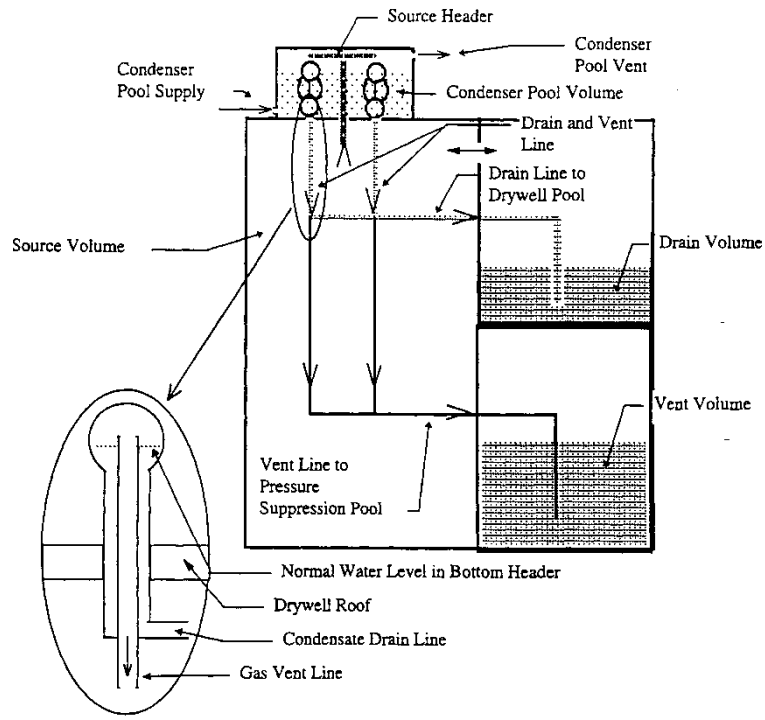


Figure 1.2 Example Flow Diagram of the Passive Containment Cooling System

The condenser model has been written to allow the user to define PCCS or ICS systems with up to three identical PCCS or ICS units. Though these units in these systems are identical, the performance of individual units may differ because they may reject heat to different pools.

In addition, multiple PCCS or ICS systems may be defined by the user. These systems may be defined with completely independent properties. There is no limit to the number of systems that can be defined by the user.

2 Input Requirements

2.1 MELGEN Input

This section provides input requirements for the CND package and includes a short description of the input quantities, their units, and default values, if any. Further description of the input variables and their meaning in the models can be found in the CND Package Reference Manual.

Input for the CND model must be inside the data block for the ESF package, which begins with the character string ESF_INPUT.

CND_PCCS_ID – PCCS System Name and Sequence Number

Required.

This record defines the unique name of the PCCS system and a user-defined sequence number of the control volume.

(1) PCCSNAME

User-defined PCCS name.

(type = character*16, default = none)

(2) IPCCSNUM

User-defined PCCS sequence number.
(type = integer, default = none)

Examples

```
CND_PCCS_ID  PCCS1  10
```

CND_PCCS01 – CVH Volumes that Interface with the PCCS

Required for Activation of the PCCS Condenser Model

This record identifies the CVH volumes that represent (1) the heat sink for the heat exchangers (a total of three pools), (2) the volume from which material is removed (drywell), (3) the volume containing the vent (wetwell), and (4) the volume containing the drain (gravity-driven cooling system (GDCS) pool).

(1) IVPCPL(1)*

CVH volume name for the primary PCCS unit pool.

(type = character*16, default = none)

(2) IVPCPL(2)*

CVH volume name for the secondary PCCS unit pool.

(type = character*16, default = none)

(3) IVPCPL(3) *

CVH volume name for the third PCCS unit pool.

(type = character*16, default = none)

(4) IVPCSO

CVH volume name representing the drywell, i.e., the source for the PCCS units.

(type = character*16, default = none)

(5) IVPCVN

CVH volume name representing the wetwell, i.e., the PCCS vent.

(type = character*16, default = none)

(6) IVPCDN

CVH volume name representing the GDCS pool(s), i.e., the PCCS drain.

(type = character*16, default = none)

Initialization Characteristics of the PCCS Model

This input record acts as a flag to activate the PCCS model. If entered, then all remaining PCCS model input cards are processed. If the input record card is absent, no memory is allocated for the PCCS model variables and the plot variables unique to the PCCS model are not defined.

Example

```
CND_PCCS01 CVol400   CVol401   CVol402   CVol210 CVol230 CVol240
```

CND_PCCS02 – Tabular Function for the Variation of Capacity

Required

---

* A CVH volume name should be input for each of the three installed PCCS units. However, the same CVH volume name may be input for one or all of the PCCS unit pools.

This record provides the input for the variation in capacity caused by various containment conditions that would affect the operation of the PCCS heat exchanger. This does not include any effects of filling the condenser volume with noncondensible gas (i.e., bounding) because the model calculates these effects.

(1) IPCDPR

Tabular Function name that represents the variation in capacity due to the increase in differential pressure between the drywell (the volume from which the PCCS removes material) and the wetwell (the vent volume for the PCCS).

(type = character*16, default = none)

(2) IPLTMP

Tabular Function name for the variation in capacity caused by an increased noncondensible mole fraction in the PCCS for an ICS/PCCS pool temperature of 323.16K (50°C).

(type = character*16, default = none)

(3) IPCNCN

Tabular Function name for the variation in capacity caused by an increased noncondensible mole fraction in the PCCS for an ICS/PCCS pool temperature of 373.16K (100°C).

(type = character*16, default = none)

(4) IPSRPR

Tabular Function name for the variation in capacity caused by decreased pressure in the PCCS source volume.

(type = character*16, default = none)

Note: The Tabular Function IPCDPR should consist of (1) differential pressures between the drywell and the wetwell and (2) the corresponding coefficients that are used as multipliers to the maximum capacity. These coefficients are applied to obtain the adjusted PCCS capacities. See CND Package Reference Manual Table 2.1.

The Tabular Functions IPLTMP and IPCNCN should consist of noncondensible gas mole fractions and the corresponding coefficients for ICS/PCCS pool temperatures of 50°C and 100°C, respectively. See CND Package Reference Manual Tables 2.2 and 2.3.

The Tabular Function IPSRPR should consist of (1) the pressure in the drywell and (2) the corresponding coefficients that are used as multipliers to the maximum capacity. These coefficients are applied to obtain the adjusted PCCS capacities. See CND Package Reference Manual Table 2.4.

Example

```
CND_PCCS02   TF300        TF301       TF302     TF303
```

CND_PCCS03 – Geometric Input for the PCCS Vent Line

Required

This record provides the input for the geometric configuration of the PCCS vent line.

(1) ELBTVT

Elevation of the bottom of the vent line, assumed to be submerged in the volume normally occupied by the pressure suppression pool.

(type = real, default = none, units = m)

(2) VNTLND

Diameter of the vent line.

(type = real, default = none, units = m)

(3) VNTLEN

Total equivalent length of the PCCS vent line. No input is allowed for elbows. Therefore, a total equivalent length should be used in place of the actual length. (e.g., Crane Technical Paper #410).

(type = real, default = none, units = m)

Example

```
CND_PCCS03   9.4   0.25   62.7
```

CND_PCCS04 – PCCS Unit(s) Description

Required

This record provides input to describe the PCCS unit and the number of units that are in operation. A PCCS unit is defined as two heat exchangers with a single source line coming from the drywell and a combined exit line. The combined exit divides outside of the PCCS enclosure into separate drain and vent lines returning to the GDCS pool and to the wetwell, respectively. If multiple units are in operation, they work in parallel, with separate but identical connecting lines.

(1) VOLPCS

Volume of one PCCS unit. It is assumed that the source line and the four barrels, located at the top and bottom of the tube bundles, are sufficiently insulated to preclude significant heat transfer from the steam to the ICS/PCCS pool. Therefore, the input volume should be limited to the volume inside the two sets of tube bundles.

(type = real, default = none, units = m^3)

(2) PCMXPF

Maximum capacity of a single PCCS unit, two heat exchangers, when filled with saturated steam. This capacity should correspond to the conditions for which the tabular functions provide multipliers of 1.00. The input record CND_PCCS02 identifies those tabular functions.

(type = real, default = none, units = MW)

(3) NMPCCS

The maximum number of PCCS units in operation is three. If a negative value is input, then two special data files are produced (see section 2 of reference manual). The heat loads for the PCCS units in operation are always assumed to be distributed evenly between the ICS/PCCS pools. The heat loads correspond to the number of units in operation starting with IVPCPL(1).

(type = integer, default = none, units = none)

(4) VLPCSL

Volume of one PCCS unit source line-composed of the two barrels located at the top of the tube bundles-along with the source line and the header. This volume is used to determine the minimum delta P between the PCCS and the drywell. This volume causes the contents of the condenser tube to return to the drywell.

(type = real, default = none, units = m^3)

Example

```
CND_PCCS04   1.6   10.6   3   2.0
```

CND_ICS_ID – ICS System Name and Sequence Number

Required.

This record defines the unique name of the ICS system and a user-defined sequence number of the control volume.

(1) ICSNAME

User-defined ICS name.

(type = character*16, default = none)

(2) ICSNUM

User-defined ICS sequence number.

(type = integer, default = none)

Examples

```
CND_ICS_ID  ICS1  10
```

CND_ICS01 – CVH Volumes that Interface with the ICS

Required for Activation of the ICS Condenser Model

This record identifies the CVH volumes that represent (1) the heat sink for the heat exchangers (a total of three pools), (2) the volume from which material is removed (reactor vessel steam dome), (3) the volume containing the vent (wetwell), and (4) the volume containing the drain (reactor vessel annulus).

(1) IVICPL(1)[†]

CVH volume name for the primary ICS unit pool.

(type = character*16, default = none)

(2) IVICPL(2)[†]

CVH volume name for the secondary ICS unit pool.

(type = character*16, default = none)

(3) IVICPL(3)[†]

CVH volume name for the third ICS unit pool.

(type = character*16, default = none)

(4) IVICSO

CVH volume name representing the reactor vessel upper steam dome, i.e., the source for the ICS units.

(type = character*16, default = none)

(5) IVICVN

CVH volume name representing the wetwell, i.e., the ICS vent.

(type = character*16, default = none)

(6) IVICDN

CVH volume name representing the reactor downcomer, i.e., the ICS drain.

(type = character*16, default = none)

Initialization Characteristics of the ICS Model

This input record acts as a flag to activate the ICS model. If this input card is entered, then all remaining ICS model input cards are processed. If the input record card is absent, no memory is allocated for the ICS model variables and the plot variables unique to the ICS model are not defined.

[†] A CVH volume name should be input for each of the three installed ICS units. However, the same CVH volume name may be input for one or all of the ICS unit pools.

Example

```
CND_ICS01  CV210  CV211  CV212  CV701  CV213  CV214
```

CND_ICS02 – Tabular Function for the Variation of ICS Capacity

Required

This record provides input for the variation in capacity caused by four containment conditions that affect operation of the ICS heat exchanger. This does not include any effects from filling the condenser volume with noncondensible gas (i.e., bounding) because the model calculates these effects.

(1) IICDPR

Tabular Function name that represents the variation in capacity due to the increase in differential pressure between the reactor vessel (the volume from which the ICS removes material) and the wetwell (the vent volume for the ICS).

(type = character*16, default = none)

(2) IILTMP

Tabular Function name for the variation in capacity because of increase in noncondensible mole fraction in the ICS for an ICS/PCCS pool temperature of 323.16K (50°C).

(type = character*16, default = none)

(3) IICNCN

Tabular Function name for the variation in capacity caused by an increased noncondensible mole fraction in the ICS for an ICS/PCCS pool temperature of 373.16K (100°C).

(type = character*16, default = none)

(4) IISRPR

Tabular Function name for the variation in capacity caused by decreased pressure in the ICS source volume.

(type = character*16, default = none)

Note: The Tabular Function IICDPR should consist of (1) differential pressures between the reactor vessel and the wetwell and (2) the corresponding coefficients that are used as multipliers to the maximum capacity. These coefficients are applied to obtain the adjusted ICS capacities.

The Tabular Functions IILTMP and IICNCN should consist of noncondensible gas mole fractions and the corresponding coefficients for ICS/PCCS pool temperatures of 50°C and 100°C, respectively.

The Tabular Function IISRPR should consist of (1) the pressure in the reactor vessel and (2) the corresponding coefficients that are used as multipliers to the maximum capacity. These coefficients are applied to obtain the adjusted ICS capacities. See CND Package Reference Manual Table 3.1.

Example

```
CND_ICS02  TF10  TF11  TF12  TF14
```

CND_ICS03 – Geometric Input for the ICS Vent Line

Required

This record provides the input for the geometric configuration of the ICS vent line.

(1) ELVICT

Elevation of the bottom of the vent line, assumed to be submerged in the volume normally occupied by the pressure suppression pool.

(type = real, default = none, units = m)

(2) VNTICD

Diameter of the vent line.

(type = real, default = none, units = m)

(3) VNTICL

Total equivalent length of the ICS vent line. No input is allowed for elbows. Therefore, a total equivalent length should be used in place of the actual length, as explained (e.g., Crane Technical Paper #410.).

(type = real, default = none, units = m)

(4) ICCFL2

Control function name that controls the operation of the ICS. A control function value other than zero simulates opening of the drain line valves that activate the ICS model.

(type = character*16, default = none)

Example

```
CND_ICS03  3.0  0.01885  17.4  CF888
```

CND_ICS04 – ICS Unit(s) Description

Required

This record provides input to describe the ICS unit and the number of units that are in operation. An ICS unit is defined as two heat exchangers with a single source line coming from the reactor vessel, two drain lines that merge before returning to the reactor vessel downcomer, and two vent lines that merge before returning to the wetwell. If multiple units are in operation, they work in parallel, with separate but identical connecting lines.

(1) VOLIC

Volume of one ICS unit. It is assumed that the source line and the four barrels-located at the top and bottom of the tube bundles- are sufficiently insulated to preclude significant heat transfer from the steam to the ICS/PCCS pool. Therefore, the input volume should be limited to the volume inside the two sets of tube bundles.

(type = real, default = none, units = m^3)

(2) PFICMX

Maximum capacity of a single ICS unit consisting of two heat exchangers, when filled with saturated steam. This capacity should correspond to the conditions for which tabular functions provide multipliers of 1.00. The tabular functions are identified in input record CND_ICS02.

(type = real, default = none, units = MW)

(3) NMIC

The maximum number of ICS units that are currently in operation, is three. If a negative value is input, then two special data files are produced (see section 3 of reference manual). The heat loads for the ICS units in operation are always assumed to be distributed evenly between the ICS/PCCS pools. These heat loads correspond to the number of units in operation starting with IVICPL(1).

(type = integer, default = none, units = none)

(4) VNSTP(1)

Reactor vessel pressure at which the vent line valves open.

(type = real, default = none, units = Pa)

(5) VNSTP(2)

Reactor vessel pressure reset point for the ICS vent line valves (pressure at which the valves close).

(type = real, default = none, units = Pa)

(6) VNSTP(3)

Minimum time delay before the vent line valves are allowed to close. A control function may also be used to control the operation of the ICS vent valve by making VNSTP(3) the name of the control function.

(type = real or character*16, default = none, units = s)

(7) VLICSL

Volume of one ICS unit source line-composed of the two barrels located at the top of the tube bundles-along with the source line and the steam header. This volume is used to determine the minimum delta P between the ICS and the reactor vessel steam dome. This volume causes the contents of the condenser tubes to return to the reactor vessel.

(type = real, default = none, units = m^3)

Example

```
CND_ICS04  0.7368  30.0  3  7.653e6  7.584e6  240.0  2.986
```

2.2 MELCOR Input

Input records CND_PCCS04 and CND_ICS04, defined in the previous sections for MELGEN, may also be included in MELCOR input but are not required. If input, then all values must be present. In effect, insertion of this record allows the user to modify the number of PCCS or ICS units in operation on MELCOR restarts. However, the number of units in operation is limited to the number of ICS/PCCS pools defined in input record CND_PCCS01 for the PCCS and record CND_ICS01 for the ICS. Also, the creation of the external data files can be controlled by changing the sign of the NMPCCS or NMIC variable in the MELCOR input upon restart.

3 Special Output Data Files

The CND package can produce detailed performance-data files. See Table 3.1. The file names are currently defined within the code and may be unacceptable for some operating systems.

If a negative value is input for NMPCCS (record CND_PCCS04), then two special data files are produced by the PCCS model routine CNDRNI. The first data file is named PLOT.DAT. The second is PLOTSPECIAL.DAT. These data files include detailed information on the performance of the PCCS units that are in operation for every timestep. In both files, the time is printed in Column 1.

The PLOT.DAT file contains the overall performance characteristics of the PCCS units that are in operation. The second column is the amount of power transferred to the ICS/PCCS pools, E2ADIC. The amount of power transferred to each individual pool can be determined by dividing E2ADIC by the number of PCCS units in operation. The third column is the maximum capacity that can be used if the PCCS has not become blocked, i.e., PLTCEF. The difference between PLTCEF and E2ADIC is the amount of power that could not be utilized during the current timestep because of noncondensible gas blocking of the PCCS.

The fourth column provides the total flow of material from either the drywell or the PCCS through the vent line to the wetwell, PLTIFL. The fifth column is the amount of

noncondensible gases that flows through the vent line to the wetwell, PLTNF. The difference between PLTIFL and PLTNFL is the flow of uncondensed steam through the vent.

The sixth column, PLTVLL, is the amount of void left in the PCCS at the end of the current timestep. See CND Reference Manual for information concerning the significance of this variable. The seventh column, PRPSOU, is the noncondensible mole fraction in the source volume determined by using partial pressures. The last two variables (columns 8 and 9) are the differential pressure between the drywell and the wetwell, DELPRE; and the differential pressure required to produce flow between the drywell and the wetwell, REQPRE.

The data file PLOTSPECIAL.DAT contains information useful for detailed analysis of various internal calculations performed by the PCCS model. As before, the first column provides the time. The second column (ENCDCL) is the power used to cool any noncondensible gases, which remained in the PCCS at the end of the previous timestep, to the ICS/PCCS pool temperature.

The third and fourth columns contain the powers associated with the materials that are vented to the wetwell. The first of these variables is the power given up to the pressure suppression pool by the fog or uncondensed steam or by heating/cooling of the noncondensible gases passing through the water to the wetwell atmosphere, PS_WW_PL. The second variable is the amount of power added to the wetwell atmosphere; this is the power associated with the noncondensible gases at equilibrium with the temperature of the wetwell pool, PS_WW_ATM.

The fifth column provides the pressure in the PCCS at the beginning of the timestep after the iterative operation of filling the PCCS is performed, PRPCS_2. PRPCS_2 should be the same as the sixth variable, the pressure in the drywell, P_DRYWELL.

The four subsequent columns (7–10) provide the amount of steam before and after the call to the equilibration routine, STM_MASS_BEF and STM_MAS_AFT, and the amount of fog before and after the call to the equilibration routine, FOG_MAS_BEF and FOG_MAS_AFT.

Column 11 is the temperature of the noncondensible gases that remain in the PCCS from the end of the previous timestep, TATPCS_1. This is followed by the temperature of the PCCS gases after the call to the equilibration routine, TATPCS_2 in column 12. The temperature of the drywell, T-DRYWELL, is provided in column 13. Finally, the number of iterations performed in filling the PCCS, NCOUNT, is listed in the last column.

If a negative value is input for NMIC (record CND_ICS04) then two special data files are produced by the modeling routine CNDRN1. The first of these data files for the ICS is named PLOT.RX.DAT. The second is PLOTSPECIAL.RX.DAT. Similar to the corresponding files for the PCCS, these data files include detailed information on the performance of the ICS units that are in operation every timestep. Table 3.1 gives a short description of each variable in the four data files.

The ICS data files contain exactly the same information as described previously for the PCCS, except that the source volume is the reactor vessel instead of the drywell.

A data set is produced every time the PCCS and ICS models are called; therefore, if a fallback occurs, one or more data point sets will be produced for the same beginning time. To remove the timestep information rejected by the fallback maneuver and to improve the appearance of the data files, a separate post processor was written to change the data over to a format that could be easily placed into either a consistent plotting package or a spreadsheet package.

Table 3.1 Description of Variables in ICS/PCCS Data Files

PLOT.DAT			
	Variable Name	**Units**	**Description**
1	TIME	s	Beginning of timestep value.
2	E2ADIC	W	Power transferred to the ICS/PCCS pools.
3	PLTCEF	W	Maximum capacity of the PCCS unit(s) in operation.
4	PLTIFL	kg/s	Flow rate of material through the vent line(s).
5	PLTNFL	kg/s	Flow rate of noncondensible gases through the vent line(s).
6	PLTVLL	m^3	Void left in the PCCS at the end of the current timestep.
7	PRPSOU	(no units)	Mole fraction of noncondensible gases in source volume.
8	DELPRE	Pa	Differential pressure between the source volume (drywell) and the vent volume (wetwell).
9	REQPRE	Pa	Minimum differential pressure required to produce flow.

Table 3.2 Description of Variables in ICS/PCCS Data Files (Continue)

PLOTSPECIAL.DAT			
	Variable Name	**Units**	**Description**
1	TIME	s	Beginning of timestep value.
2	ENCDCL	W	Power used to cool any noncondensible gases that remained in the PCCS at the end of the previous timestep to the ICS/PCCS pool temperature.
3	PW_WW_PL	W	Power passed to the pressure suppression pool.
4	PW_WW_ATM	W	Power passed to the wetwell atmosphere.
5	PRPCS_2	Pa	Pressure in the PCCS after the iterative filling operation.
6	P_DRYWELL	Pa	Pressure in the drywell.
7	STM_MAS_BEF	kg	Steam mass before the equilibration of the material in the condenser.
8	STM_MAS_AFT	kg	Steam mass after the equilibration of the material in the condenser.
9	FOG_MAS_BEF	kg	Fog mass before the equilibration of the material in the condenser.
10	FOG_MAS_AFT	kg	Fog mass after the equilibration of the material in the condenser.
11	TATPCS_1	K	Temperature of noncondensible gases that remain in the PCCS from the end of the previous timestep.
12	TATPCS_2	K	Temperature of the PCCS gases after equilibration of the material in the condenser.
13	T-DRYWELL	K	Temperature of the gases in the drywell.
14	NCOUNT	(no units)	Number of iterations performed in filling the PCCS.
PLOT.RX.DAT			
	Variable Name	**Units**	**Description**
1	TIME	s	Beginning of timestep value.
2	E2ADIC	W	Power transferred to the ICS/PCCS pools.
3	PLTCEF	W	Maximum capacity of the ICS unit(s) in operation.
4	PLTIFL	kg/s	Flow rate of material through the vent line(s).
5	PLTNFL	kg/s	Flow rate of noncondensible gases through the vent line(s).
6	PLTVLL	m^3	Void left in the ICS at the end of the current timestep.
7	PRPSOU	(no units)	Mole fraction of noncondensible gases in source volume.
8	DELPRE	Pa	Differential pressure between the source volume (reactor vessel) and the vent volume (wetwell).
9	CLRPRE	Pa	Differential pressure required to clear the water column from the vent line in the wetwell pool.

Table 3.3 Description of Variables in ICS/PCCS Data Files (Continue)

PLOTSPECIAL.RX.DAT			
	Variable Name	**Units**	**Description**
1	TIME	s	Beginning of timestep value.
2	ENCDCL	W	Power used to cool any noncondensible gases that remained in the ICS at the end of the previous timestep, to the ICS/PCCS pool temperature.
3	PW_WW_PL	W	Power passed to the pressure suppression pool.
4	PW_WW_ATM	W	Power passed to the wetwell atmosphere.
5	PRIC_2	Pa	Pressure in the ICS after the iterative filling operation.
6	P_RPV	Pa	Pressure in the reactor vessel.
7	STM_MAS_BEF	kg	Steam mass before the equilibration of the material in the condenser.
8	STM_MAS_AFT	kg	Steam mass after the equilibration of the material in the condenser.
9	FOG_MASS_BEF	kg	Fog mass before the equilibration of the material in the condenser.
10	FOG_MAS_AFT	kg	Fog mass after the equilibration of the material in the condenser.
11	TATIC_1	K	Temperature of noncondensible gases that remain in the ICS from the end of the previous timestep.
12	TATIC_2	K	Temperature of the ICS gases after equilibration of the material in the condenser.
13	T-RPV	K	Temperature of the gases in the reactor vessel.
14	NCOUNT	(no units)	Number of iterations performed in filling the ICS.

4 Plot Variables

The plot variables included in the ESFCND package are listed below, along with a brief description of each.

ESF-PCCS-VNTFL.SYS — For each PCCS system (SYS), the integrated total flow through the PCCS vent line into the pressure suppression pool.

(units = kg)

ESF-PCCS-TOTENG.SYS — For each PCCS system (SYS), the integrated energy added to the ICS/PCCS pools (PCCS heat sink).

(units = J)

ESF-PCCS-TOTSTM.SYS	For each PCCS system (SYS), the cumulative steam removed from the drywell. (units = kg)
ESF- PCCS -POW	Power transferred to the ICS/PCCS pools PCCS heat sink). (units = W)
ESF- PCCS -STM	Steam removal rate from the drywell. (units = kg/s)
ESF- PCCS -COND	Condensate return flowrate from the ICS to the reactor vessel. (units = kg/s)
ESF- PCCS -TCOND	Temperature of the condensate returned from the PCCS. (units = K)
ESF-ICS-VNTFL.SYS	For each ICS system (SYS), the integrated total flow through the ICS vent line into the pressure suppression pool. (units = kg)
ESF-ICS-TOTENG.SYS	For each ICS system (SYS), the integrated energy added to the ICS/PCCS pools (ICS heat sink). (units = J)
ESF-ICS-TOTSTM.SYS	For each ICS system (SYS), the cumulative steam removed from the reactor vessel. (units = kg)
ESF-ICS-POW	Power transferred to the ICS/PCCS pools (ICS heat sink). (units = W)
ESF-ICS-STM	Steam removal rate from the reactor vessel. (units = kg/s)
ESF-ICS-COND	Condensate return flowrate from the ICS to the reactor vessel. (units = kg/s)
ESF-ICS-TCOND	Temperature of the condensate returned from the ICS to the reactor vessel.

(units = K)

5 Control Function Arguments

The control function arguments are described in this section. Note that plot variables (some are identical in definition to these control function arguments but different in format) are described in the previous section.

ESF-ICS-TOTENG	Integrated energy added to the ICS/PCCS pools (ICS heat sink). (units = J)
ESF-ICS-POW	Power transferred to the ICS/PCCS pools (ICS heat sink). (units = W)
ESF-ICS-STM	Steam removal rate from the reactor vessel. (units = kg/s)
ESF-ICS-COND	Condensate return flowrate from the ICS to the reactor vessel. (units = kg/s)

6 Example Input

The following are sample MELGEN input records for the CND package and associated control functions. An example of MELCOR input deck follows.

6.1 Example MELGEN Input

```
!* Block: ESF data
!**************************************************
ESF_INPUT
!
CND_PCCS_ID PCCS1   110  !Multiple PCCS systems could be defined.
!*************************************************************
!
! Ivpcpl – cvh volumes receiving heat transfer from PCCS
! Ivpcso – cvh volume from which the source is taken for the
! PCCS
! Ivpcvn – cvh volume containing the vent line from the PCCS
! Ivpcdn – cvh volume containing the drain line from the PCCS
!        ivpcpl(1) ivpcpl(2) ivpcpl(3) ivpcso    ivpcvn   ivpcdn
CND_PCCS01 CV400    CV401     CV402    CVH210    CV230    CV240
!
! ipcdpr - variation in efficiency due to difference in
! differential pressure between the drywell and wetwell
```

```
! ipltmp - variation in efficiency due to increase of
! noncondensible partial pressure in the pccs for a pool
! temperature of 50 degree C
! ipcncn - variation in efficiency due to increase of
! noncondensible partial pressure in the pccs for a pool
! temperature of 100 degree C
! ipsrpr - variation in efficiency due to decrease in source
! pressure
!            ipcdpr        ipltmp       ipcncn         ipsrpr
CND_PCCS02 'delp rduct pccs' 'nncn rd 50 pccs' 'nncn rd 100 pccs' 'PCCS Source
pres'
! elbtvt – elevation of the bottom of the vent line from pccs
! vntlnd – the minimum diameter of the vent line in meters
! vntlen – the vent line length in meters
!            elbtvt   vntlnd  vntlen
CND_PCCS03   9.350   .25451   62.67
! volpcs – volume of one unit of the pccs in m**3
! pcmxpf – maximum performance at design operating conditions MWt
! nmpccs – number of pccs units operating
! vlpcsl – volume of one unit of the pccs source line in m**3
!           volpcs pcmxpf nmpccs vlpcsl
CND_PCCS04   1.58   10.6    3    2.01
!*****************************************************************
!
CND_ICS_ID ICS1   5  !  Multiple Isolation Condenser systems can be defined
! Ivicpl – cvh volumes receiving heat transfer from ICS
! Ivicso – cvh volume from which the source is taken for the ICS
! Ivicvn – cvh volume containing the vent line from the ICS
! Ivicdn – cvh volume containing the drain line from the ICS
!*        ivicpl(1) ivicpl(2) ivicpl(3) ivicso    ivicvn    ivicdn
CND_ICS01 CV300     CV301     CV302     CV160     CV230     CV100
! iicdpr – variation in efficiency due to difference in diffential
! pressure between the reactor vessel and
! wetwell
! iiltmp - variation in efficiency due to increase of
! noncondensible partial pressure in the ics
! for a pool temperature of 50 degree C
! iicncn - variation in efficiency due to increase of
! noncondensible partial pressure in the ics
! for pool temperature of 100 degree C
! ipsrpr - variation in effiency due to decrease
! in source pressure
!         iicdpr         iiltmp       iicncn         iisrpr
CND_ICS02 'delp rduct ics' 'nncn rd 50 ics' 'nncn rd 100 pccs' 'ICS Source pres'
!
! Values for IICDPR, IILTMP, IICNCN, have not been obtained from
! GE, thus the same values for the PCCS are used for the ICS
!
! elvict – elevation of the bottom of the vent line from ics
! vnticd – the minimum diameter of the vent line in meters
! vnticl – the vent line length in meters
! iccfl2 - control function indicating initiation of the ICS,
! a non-zero number initiates operation
! the value can be + or –
!         elvict   vnticd  vnticl  iccf12
CND_ICS03   9.5   0.01885   17.4   CF805
!
```

```
! volic - volume of one unit of the ics in m**3
! pficmx - maximum performance at design operating conditions MWt
! nmic - number of ics units operating
! vnstp(1) opening reactor vessel pressure for ics vent Pa
! vnstp(2) reset reactor vessel pressure for ics vent Pa
! vnstp(3) time delay before which the vent can not close
! if vnstp(3) is negative then a CF value must be
! provided that controls the function of the vent valve
! vlicsl - volume of one unit of the ics source line in m**3
!           volic   pficmx nmic vnstp(1) vnstp(2) vnstp(3) vlicsl
CND_ICS04   0.7368   30.0   3   7.653e6   7.584e6   240.0   2.986
!
!*************************************************************
!
TF_INPUT
!*************************************************************
!          TFNAME       TFSCAL  TFADCN
TF_ID 'PCCS Source pres'   1.0   0.0
TF_TAB 14 ! NTFPAR     X       Y
!
! Pressure Multiplier to PCCS Pa Performance
!
            1      0.00E+00 0.0000
            2      6.11E+02 0.0000
            3      5.00E+04 0.0000
            4      1.00E+05 0.0000
! For pressures below .15 MPa the saturation temperature is
! below
! 100 C; therefore, no heat transfer is allowed
            5      1.50E+05 0.4250
            6      2.00E+05 0.6660
            7      2.50E+05 0.8495
! Base operating pressure is approx. 0.3 MPa
            8      3.00E+05 1.0000
! at higher pressure the steam is more dense; therefore, the
! pccs becomes more effective
            9      3.50E+05 1.1289
            10     4.00E+05 1.2425
            11     4.50E+05 1.3450
            12     5.00E+05 1.4386
            13     6.50E+05 1.6807
            14     7.00E+05 1.7518
!
! The base case is chosen to be the PCCS at the following
! conditions:
! IC/PCCS pool Temperature = 100 C
! Delta P = 15.4 Kpa
! a NC Mole Fraction of 0.0
! and a base capacity of 10.6 MWt
!*
! values from ge for a typical pccs unit
TF_ID ‘delp rduct pccs'   1.0   0.0
TF_TAB 5 ! NTFPAR     X       Y
! pa – verification in capacity
            1      0.0     0.9434
            2      7500.0  0.9434
            3      15400.  1.00
```

```
            4      23400.0 1.0094
            5      1.0e+06 1.0094
! values from ge for a ic/pccs pool
! temperature of 50 c
TF_ID 'nncn rd 50 pccs'   1.0   0.0
TF_TAB 21 ! NTFPAR     X       Y
! mole fraction of noncondensibles – variation in cap.
            1      0.00   2.2547
            2      0.0071 1.5094
            3      0.0287 1.3302
            4      0.0570 1.1981
            5      0.1142 1.0377
            6      0.1718 0.9085
            7      0.2290 0.7925
            8      0.2861 0.6887
            9      0.3429 0.5962
            10     0.3999 0.5123
            11     0.4571 0.4368
            12     0.5140 0.3689
            13     0.5718 0.3075
            14     0.6289 0.2509
            15     0.6852 0.1981
            16     0.7429 0.1491
            17     0.8006 0.1028
            18     0.8568 0.0620
            19     0.9141 0.0266
            20     0.9708 0.0012
            21     1.0000 0.0000
!
! values from ge for a ic/pccs pool temperature of 100 c
TF_ID 'nncn rd 100 pccs'   1.0   0.0
TF_TAB 16 ! NTFPAR     X       Y
! mole fraction of noncondensibles – variation in cap.
            1      0.00   1.0000
            2      0.0071 0.9434
            3      0.0287 0.7726
            4      0.0570 0.6858
            5      0.1142 0.5811
            6      0.1718 0.4972
            7      0.2290 0.4255
            8      0.2861 0.3585
            9      0.3429 0.2962
            10     0.3999 0.2396
            11     0.4571 0.1868
            12     0.5140 0.1387
            13     0.5718 0.0926
            14     0.6289 0.0514
            15     0.6852 0.0152
            16     1.0    0.0000
!
! variation values from ge for a ic/pccs pool
! temperature of 50 c obtained from PCCS data. Upper value
! limited to 1.5 for a mole fraction of 0.0
TF_ID 'nncn rd 50 ics'   1.0   0.0
TF_TAB 21 ! NTFPAR     X       Y
! mole fraction of noncondensibles – variation in cap.
            1      0.00     1.5
```

```
            2      0.0071   1.5094
            3      0.0287   1.3302
            4      0.0570   1.1981
            5      0.1142   1.0377
            6      0.1718   0.9085
            7      0.2290   0.7925
            8      0.2861   0.6887
            9      0.3429   0.5962
            10     0.3999   0.5123
            11     0.4571   0.4368
            12     0.5140   0.3689
            13     0.5718   0.3075
            14     0.6289   0.2509
            15     0.6852   0.1981
            16     0.7429   0.1491
            17     0.8006   0.1028
            18     0.8568   0.0000
            19     0.9141   0.0266
            20     0.9708   0.0012
            21     1.0000   0.0000

TF_ID 'ICS Source pres'   1.0   0.0
TF_TAB 46 ! NTFPAR     X       Y
! Pressure Multiplier to ICS Pa Performance
            1      0.00E+00 0.0000
            2      6.11E+02 0.0000
            3      5.00E+04 0.0000
            4      1.00E+05 0.0000
! For pressures below .15 MPa the saturation temperature is below
! 100 C; therefore, no heat transfer is allowed
            5      1.50E+05 0.1080
            6      2.00E+05 0.1692
            7      2.50E+05 0.2159
            8      3.00E+05 0.2541
            9      3.50E+05 0.2869
            10     4.00E+05 0.3157
            11     4.50E+05 0.3418
            12     5.00E+05 0.3655
            13     6.50E+05 0.4271
            14     7.00E+05 0.4451
            15     7.50E+05 0.4624
            16     8.00E+05 0.4787
            17     8.50E+05 0.4943
            18     9.00E+05 0.5092
            19     9.50E+05 0.5237
            20     1.00E+06 0.5376
            21     1.10E+06 0.5643
            22     1.20E+06 0.6507
            23     1.30E+06 0.6139
            24     1.40E+06 0.6368
            25     1.50E+06 0.6591
            26     1.75E+06 0.6661
            27     2.00E+06 0.6980
            28     2.25E+06 0.7259
            29     2.50E+06 0.7512
            30     3.00E+06 0.7956
            31     3.50E+06 0.8328
```

```
            32     4.00E+06 0.8645
            33     5.00E+06 0.9159
            34     6.00E+06 0.9556
            35     7.00E+06 0.9883
            36     7.20E+06 0.9942
! Base operating pressure is approx. 7.4 MPa
            37     7.40E+06 1.0000
            38     7.60E+06 1.0054
            39     7.80E+06 1.0106
            40     8.00E+06 1.0156
            41     8.20E+06 1.0204
            42     8.40E+06 1.0252
            43     8.60E+06 1.0298
            44     8.80E+06 1.0340
            45     9.00E+06 1.0381
            46     1.00E+07 1.0568

! no variation in performance is assumed for increase delta p in
! the vent line
TF_ID 'delp rduct ics'   1.0   0.0
TF_TAB 2! NTFPAR     X       Y
! pa – variation in capacity
            1      0.0      1.00
            2      1.0e+06  1.00
!*************************************************************
CF_INPUT
!
!*************************************************************
!
!* The following cf's are an example of the use of the CF logic
!*
!* to approximate the ics vent valve operation
!*****************************************************************
!
!*****************************************************************
!
CF_ID  'CF301' L-GE
CF_LIV TRUE
CF_ARG    2 ! NARG CHARG       ARSCAL   ARADCN
                1 EXEC-TIME     1.0      100.0 ! 100 + current time
                2 EXEC-TIME     0.0       0.0  ! zero

CF_ID  'CF302' L-GE
CF_LIV FALSE
CF_ARG  2 ! NARG CHARG   ARSCAL ARADCN
             1 EXEC-TIME  –1.0   0.0 ! negative of the current time
             2 EXEC-TIME   1.0 100.0 ! 100 + current time

CF_ID  'CF303' L-GE
CF_LIV FALSE
CF_ARG  2 ! NARG CHARG   ARSCAL ARADCN
             1 CVH-P(CV160) 1.0   0.0  !pressure in reactor vessel,
                                       ! CVH volume name CV160
             2 EXEC-TIME   0.0 7.65e06 ! upper set point of vent
                                       ! valve

CF_ID  'CF304' L-GE
```

```
CF_LIV TRUE
CF_ARG  2 ! NARG CHARG     ARSCAL ARADCN
             1 CVH-P(CV160) –1.0   0.0  !pressure in reactor vessel
             2 EXEC-TIME   0.0 -7.584e06 ! lower set point for vent
                                         ! valve closure

CF_ID  'CF305' L-NOT
CF_LIV FALSE
CF_ARG  1 ! NARG CHARG       ARSCAL  ARADCN
             1 CF-VALU(CF313)  1.0    0.0

CF_ID  'CF306' L-AND
CF_LIV FALSE
CF_ARG  2 ! NARG CHARG     ARSCAL ARADCN
             1 CF-VALU(CF303)  1.0    0.0
             2 CF-VALU(CF305)  1.0    0.0

CF_ID 'CF307' L-L-IFTE
CF_LIV FALSE
CF_ARG    3 ! NARG CHARG           ARSCAL  ARADCN
                1 CF-VALU(CF306)    1.0     0.0 ! check to see if
                                        ! p>set point and .not.timer
                2 CF-VALU(CF301)    1.0     0.0 ! set the timer
                                                ! equal to true
                3 CF-VALU(CF313)    1.0     0.0 ! set the timer
                                        ! equal to its old value

CF_ID 'CF308' L-A-IFTE
CF_SAI   1.0   0.0   0.0
CF_ARG    3 ! NARG CHARG          ARSCAL ARADCN
                1 CF-VALU(CF306)   1.0    0.0 ! check to see if
                                    ! p>set point and .not.timer
                2 EXEC-TIME        1.0    60.0 ! set the closure
                                    ! time to current time + 60s
                3 CF-VALU(CF308)   1.0    0.0 ! set the closure
                                    ! time to the old value

CF_ID 'CF309' L-A-IFTE
CF_SAI   1.0   0.0   0.0
CF_ARG    3 ! NARG CHARG          ARSCAL ARADCN
                1 CF-VALU(CF303)   1.0    0.0 ! check to see if
                                              ! p>set point
                2 EXEC-TIME        0.0    1.0 ! set the valve
                                              ! opening to 1.0
                3 CF-VALU(CF312)   1.0    0.0 ! set the opening to
                                              ! the old value
!
CF_ID  'CF310' L-GE
CF_LIV TRUE
CF_ARG  2 ! NARG CHARG      ARSCAL ARADCN
             1 EXEC-TIME      1.0    0.0 ! current time
             2 CF-VALU(CF308) 1.0    0.0 ! time valve can close

CF_ID  'CF311' L-AND
CF_LIV TRUE
CF_ARG  2 ! NARG CHARG      ARSCAL ARADCN
             1 CF-VALU(CF304) 1.0    0.0 ! true if pressure has
```

```
                                          ! dropped
             2 CF-VALU(CF310) 1.0    0.0 ! true if timer has passed
                                          ! current time

CF_ID 'CF312' L-A-IFTE
CF_SAI   1.0   0.0   0.0
CF_ARG    3 ! NARG CHARG          ARSCAL ARADCN
                1 CF-VALU(CF311)   1.0    0.0 ! if true close valve
                2 EXEC-TIME        0.0    0.0 ! value of zero
                3 CF-VALU(CF309)   1.0    0.0 ! valve is open set
                                             ! to the old value

CF_ID 'CF307' L-L-IFTE
CF_LIV FALSE
CF_ARG    3 ! NARG CHARG           ARSCAL  ARADCN
                1 CF-VALU(CF311)    1.0     0.0 ! if true close
                                                ! valve
                2 CF-VALU(CF302)    1.0     0.0
                3 CF-VALU(CF307)    1.0     0.0 ! valve is open set
                                                ! to the old value
!*
!*****************************************************************
!
!*****************************************************************
!
! TRIP FOR RPV WATER LEVEL 2
!*****************************************************************
!
!*****************************************************************
!
!
! The following are the control functions that are used to initiate
! operation of the ICS
!
! >>>>
! First, test to see if the collapsed water level has fallen to L2.
! >>>>
!
CF_ID 'CF803' L-GE
CF_LIV TRUE
CF_CLS LATCH ! Latch False When water level drops below L2
CF_MSG FULL-OUTPUT 'Collapsed Water Level Has Dropped Below L2'
CF_ARG    2 ! NARG CHARG              ARSCAL   ARADCN
                1 CVH-CLIQLEV(CV100)    1.0      0.0 ! collapsed
                                         ! liquid level in annulus
                2 EXEC-TIME             0.0     13.593
!
! >>>>
! Second, trap and hold the time at which the level reaches L2
! >>>>
! Following function will trip-on-reverse when CF800 fires, yielding
! a negative value, the absolute value of which is the time (s)
! since trip.
!

CF_ID 'CF804' TRIP
CF_SAI    1.  0.  0.
```

```
CF_ARG    1 ! NARG CHARG            ARSCAL ARADCN
                1 CF-VALU(CF803)      1.0   0.0
! >>>>
! Third, trap and hold the absolute value of CF804. This will be
the
! actual time since the vessel water level dropped to L2.
! >>>>
CF_ID 'CF805' ABS
CF_SAI    1.  0.  0.
CF_ARG    1 ! NARG CHARG            ARSCAL ARADCN
                1 CF-VALU(CF804)      1.0   0.0
Example MELCOR Input
!*****************************************************************
ESF_INPUT
!
! PCCS Input
! decrease the number of pccs in operation from 3 to 2
!
! volpcs - volume of one unit of the pccs in m**3
! pcmxpf - maximum performance at design operating conditions MWt
! nmpccs - number of pccs units operating
! vlpcsl - volume of one unit of the pccs source line in m**3
!         volpcs pcmxpf nmpccs vlpcsl
CND_PCCS04 1.58    10.6    2    2.01
!
! ICS Input
! decrease the number of ics in operation from 3 to 1
! also start to use the control function logic to control
! the operation of the vent valves
!
! volic - volume of one unit of the ics in m**3
! pficmx - maximum performance at design operating conditions MWt
! nmic - number of ics units operating
! vnstp(1) opening reactor vessel pressure for ics vent Pa
! vnstp(3) time delay before which the vent can not close if
! vnstp(3) is negative then a CF value must be
! provided that controls the function of the vent valve
! vlicsl - volume of one unit of the ics source line in m**3
!            volic pficmx nmic  vnstp(1)  vnstp(2) vnstp(3) vlicsl
CND_ICS04   0.7368   30.0   1   7.653e6   7.584e6   -311    2.986
```

7 CND Model Output

In general, the CND model output is self-explanatory. The integrated mass transfer through the vent to the CVH volume—which includes the suppression pool, the integrated energy transfer to the condenser pools, and the integrated mass of steam removed from the source—is output for each condenser type.

8 Diagnostic and Error Messages

Diagnostic messages will be written by MELGEN to report errors or inconsistencies in input.

Typical errors include errors in record format and failure to supply all required input records. Inconsistencies between input to different packages are also identified; each of the volumes that a condenser connects to must be defined; any tabular functions or control functions referred to by the condenser model must be properly defined.

No restart file will be written until all errors identified during input processing have been corrected. This does not, of course, assure that the accepted input properly describes the physical system that the user intended to model. The analyst should always examine the initial edit produced by MELGEN prior to running MELCOR.

Messages from MELCOR report problems encountered during the time advancement of a calculation. If an error is severe, the calculation will be stopped and a restart dump written corresponding to conditions at the beginning of the fatal timestep.

Error or warning messages may be issued during execution of MELCOR for any number of reasons. These include (1) an attempt by the Condenser package to subtract more mass of energy than is contained in the associated CVH control volume, (2) an attempt to add more mass or energy to a CVH volume than is allowed, and (3) errors found in tabular or control functions. Execution will continue, with a reduced system timestep, when it makes sense to do so; the request for a system fallback will be noted with one of the following messages:

```
TIMESTEP CUT IN CND PACKAGE
TOO LARGE A MASS SINK GENERATED
TIMESTEP CUT IN CND PACKAGE
TOO LARGE AN ENERGY SINK GENERATED
TOO LARGE AN ENERGY SINK GENERATED IN ATMOSPHERE
TIMESTEP CUT IN CND PACKAGE
```

When a problem is encountered with a tabular function or a control function, one of the following messages will be produced by routine CNDRN1:

```
tabular function error in subroutine cndrn1:50C
```

An error occurred when trying to determine the capacity reduction factor, determined from a user-defined tabular function, for a pool temperature of 50°C.

```
tabular function error in subroutine cndrn1:100C
```

An error occurred when trying to determine the capacity reduction factor, determined from a user-defined tabular function, for a pool temperature of 100°C.

```
tabular function error in subroutine cndrn1:FLOW
```

An error occurred when trying to determine the capacity reduction factor, determined from a user-defined tabular function, for the differential pressure between the source and the vent CVH volumes.

```
tabular function error in subroutine cndrn1:PRES
```

An error occurred when trying to determine the capacity reduction factor, determined from a user-defined tabular function, for the pressure in the source volume.

```
tabular function error in subroutine cndrn1:vnstp
```

Once the condenser is filled from the source volume, an initial pass is made through an equilibration routine. The equilibration routine that is utilized is CVTWGE. CVTWGE is the routine used by the CVH package to equilibrate the CVH volumes when the equilibrium option is used on card CV_THR, i.e., *Tpool* = *Tatmos*. If an error occurs in evaluating the properties of the materials in the condenser, the following error message will be sent:

```
error from call to CVTWGE in subroutine cndrnl
```

In addition to this message, this error condition will result in a restart file being written and immediate termination of the calculation.

As explained in Section 2.4 of the Condenser Package Reference Manual, an iteration process is used to fill the condenser. However, to prevent an infinite loop, the logic is limited to a maximum number of iterations. If the pressure of the source volume is not obtained within the allowable limit, the following error message is produced:

```
message from subroutine cndrnl
iteration process failed
```

Two energy balance checks are performed during MELCOR execution. The first is performed to ensure that the equilibration routine conserves energy. If energy is not conserved, then the following message is produced:

```
cndrnl:internal energies out of bound
```

The second check is an overall energy balance on the package. The energy error is defined as the current energy minus the integrated energy transferred to the package plus the integrated energy transferred from the package. The relative energy error is given by the energy error divided by the sum of the current energy and the integrated energy transferred out of the package. If the relative energy error exceeds 0.5 percent, the following message is written:

```
*****************************************
Message from CNDRN2
CND Package Energy Balance Greater than 0.5%
```

Core (COR) Package
Users' Guide

The MELCOR Core (COR) package calculates the thermal response of the core and lower plenum internal structures, including the portion of the lower head directly below the core. The package also models the relocation of core and lower plenum structural materials during melting, slumping, and debris formation, including failure of the reactor vessel and ejection of debris into the reactor cavity. Energy transfer to and from the Control Volume Hydrodynamics (CVH) package and the Heat Structure (HS) package is calculated. This Users' Guide provides basic information needed to run the COR package with the rest of MELCOR, including a short discussion of the nodalization scheme and calculation framework of the package and a detailed explanation of the user input and package output for MELGEN, MELCOR, and HISPLTM. Required and optional input, sensitivity coefficients, control function arguments, plot variables, and error messages are all covered.

More detailed information on the physics models and numerical solution schemes employed in the COR package can be found in the COR Package Reference Manual.

Contents

CONTENTS **2**

LIST OF FIGURES **9**

1 INTRODUCTION **10**

1.1 Nodalization Scheme 11
1.2 Calculation Framework 15
1.3 Support of Core Structures 16
1.4 User Control of Core Oxidation and Material Relocation 18
1.5 Summary of Modeling Changes since 1.8.6 19
1.5.1 Changes in COR and Lower Head Nodalization 19
1.5.2 Core. Lower Vessel, and Lower Head Geometry in MELCOR 1.8.6 21
1.5.3 PWR Core Periphery Model 22
1.5.4 Lower Head Model 23
1.5.5 Reflood Quench Model 23
1.5.6 New Model for Release of Ag-In-Cd Control Poison 23
1.5.7 New Model for Release and Oxidation of B_4C Control Poison 24
1.5.8 User-Defined Heat Sources and Heat Transfer Paths 24
1.5.9 Arbitrary Control Function Failure of COR Components 25
1.5.10 Enhancements to Core Support Modeling 25
1.5.11 Added Flexibility in Flow Resistance (Blockage) Model 25
1.5.12 Enhanced FU Component Composition 26
1.5.13 Enhanced Flexibility in Composition of Other Components 26
1.5.14 Improved Local Temperature (dT/dz) Model 26
1.5.15 Access to Emissivity Correlations via Sensitivity Coefficients 27
1.5.16 Direct Separate Input for CN and CB Geometry 27
1.5.17 Fuel Rod Collapse Model Logic 27
1.6 Specification of Thermal Conductivity of Gap Gases 27
1.7 Use of COR Package for Special Fuel Configurations 27
1.7.1 Modeling Accident Scenarios in Spent Fuel Pools 28

2 INPUT REQUIREMENTS **30**

2.1 MELGEN General Core 31

COR_DFT – COR Package Default Scheme Record 31
COR_GP – Geometric Parameters 32
COR_RT – Reactor Type 33
COR_VP – Vessel Parameters..... 34
COR_AVP – Additional Vessel Parameters 35
COR_RF – Radiative Exchange Factors 36
COR_TP – TP Interface and Fission Power and Gap Conductance Control Functions 37
COR_CHT – Candling Heat Transfer Coefficients..... 39
COR_MS – Model Switches 40
COR_CMT – Candling Secondary Material Transport Parameters 41
COR_CCT – Component Critical Minimum Thicknesses 42
COR_BCP – Core Boundary Conduction Parameters 43
COR_DX – Global Particulate Debris Exclusion Parameters 44
COR_BK – Global Core Blockage Enhancement Factor 47
COR_NS – Global Support Rule for NS 48
COR_SS – Global Loading and Failure Rule for SS..... 50
COR_UDSS – Properties of User-Defined SS Type..... 55
COR_PC – Global Pool Heat Transfer from Bottom/Top of SS Plates 56
COR_PR – Global Downward Radiation from SS to Pool or Lower Head..... 58
COR_CR – Control Rod Silver Release 60
COR_B4C – Control Rod B_4C Oxidation Model 61
COR_ROD – Rod Collapse Model Record 62
COR_OX – PSI Air Oxidation Model 63
COR_OXB – Zircaloy Air Oxidation Breakaway model 65
COR_TKE – Core Effective Thermal Conductance..... 66
COR_PKM01 – Reactor Point Kinetics Model (1) 66
COR_PKM02 – Reactor Point Kinetics Model (2) 67
COR_PKM03 – Reactor Point Kinetics Model (3) 68
COR_TAVG - Region Average Component Temperature..... 69
COR_DIFF - Steady State Diffusion 70
COR_XPRT - Transport HTGR FP Model 71
COR_DIFT - Start HTGR FP Release Model Transient 71

2.2 MELGEN Axial Level Input..... 72

COR_ZP – Axial Level Parameters 72
COR_SHS – PWR Core Baffle (Shroud) Support Options 74

2.3 MELGEN Radial Ring Input..... 75

COR_RP – Radial Ring Parameters..... 75

2.4 MELGEN Specific Cell Input 77

COR_RBV – Cell Reference and Fluid Boundary Volumes 77
COR_CCM – Cell Component Masses 79
COR_KFU – Fuel Component Masses 81
COR_KCL – Cladding Component Masses 84
COR_KCN – Canister (CN) Component Masses 86

COR_KCB – Canister (CB) Component Masses 88
COR_KSS – Supporting Structure Component Masses..... 91
COR_KNS – Nonsupporting Structure Component Masses 93
COR_KSH – PWR Core Shroud Component Masses..... 95
COR_KFM – PWR Core Former Component Masses 97
COR_KPD – Particulate Debris Component Masses 100
COR_KPB – Particulate Debris Component Masses in Bypass..... 101
COR_KRK – SFP Rack (RK) Component Masses..... 103
COR_KRF – Reflector Component Masses 104
COR_CDM – Conglomerate Debris Masses 105
COR_CIT – Initial Component Temperatures..... 107
COR_PCT –Initial PWR Component Temperatures 109
COR_SRT – Cell Initial SFP-Rack Temperature 110
COR_EDR – Equivalent Diameter Record 111
COR_CDR – Canister Equivalent Diameter Record..... 114
COR_SDR – PWR Shroud and Former Equivalent Diameter Record 115
COR_RDR – SFP Rack Equivalent Diameter Record 116
COR_RFD – Reflector Equivalent Diameters Record 117
COR_RFG –Reflector Geometry Record 118
COR_BFA – Cell Boundary and Flow Areas Record 119
COR_SA – Surface Area Record 122
COR_CSA – Canister Surface Area Record 124
COR_SSA – PWR Shroud and Former Surface Area Record..... 125
COR_RSA – SFP Rack Surface Area Record..... 126
COR_RFA – Reflector Surface Area Record 127
COR_FCEL – Local Radiative Exchange Factor..... 128
COR_NOX – Oxidation Shutoff Control Function Record..... 130
COR_CFF – Control Function Component Failure 131
COR_QHS – Control Function Heat Source..... 132

2.5 MELGEN Lower Head Input..... 134

COR_LP – In-Vessel Falling Debris Quench Model Parameters..... 134
COR_LH – Lower Head Structure Record 135
COR_LHD – Lower Head Segments Record 136
COR_LHN – Lower Head Nodalization Record..... 138
COR_LHF – Lower Head Failure Modeling Parameters 139
COR_PEN – Lower Head Penetration Record 141

2.6 MELGEN Arbitrary Conduction or Radiation Heat Transfer Path..... 143

COR_HTR – Added COR Heat Transfer Path..... 143

2.7 MELGEN COR Material Input 146

COR_FUM – Extra Fuel Material Record 146
COR_CLM – Cladding Material Record 147
COR_CNM – Canister Material Record..... 147
COR_SHM – Core Shroud Material Record 148
COR_RFM – Reflector Material Record 148

COR_MAT – Core Material Specification Record 149
2.8 MELGEN CVH Fluid Flow Interface Input 150
COR_TIN – Hydrodynamic Control Volume Name Containing Core Cells. 150
COR_RAF – Ring Flow Area Control Functions 152
COR_FLDIR – Set flow direction by CV for dTdz model 153
2.9 MELGEN Electric Heating User Input 154
COR_ELPOW – Power and Voltage Data 155
COR_ELMAT – Heater Materials Data 156
2.10 MELGEN Edit Option Control Input 158
COR_EDV – Edit Flags Record 158
2.11 MELCOR Input 159
COR_DTC – COR Package Timestep Control Record 159
COR_TST – Physics On/Off Switches 160
COR_MSG –Enable Collapse Messaging 162
2.12 Required Input to Other MELCOR Packages 162
2.12.1 CVH, FL, and NCG Input 163
2.12.2 DCH and RN Input 163
2.12.3 HS Input 164
2.12.4 MP Input 165

3 SENSITIVITY COEFFICIENTS 165
COR_SC – COR Sensitivity Coefficients Input Record 165
3.1 Oxidation, Heat Transfer, and Relocation Parameters 166
1001 – Zircaloy Oxidation Rate Constant Coefficients 166
1002 – Steel Oxidation Rate Constant Coefficients 167
1003 – Gaseous Diffusion Oxidation Coefficients 168
1004 – Oxidation Cutoff Temperatures 168
1005 – B_4C Reaction Model Parameters 169
1006 – B_4C Reaction Rate Parameters 169
1007 – Ring Minimum Flow Area Fractions 169
1010 – Material Dissolution Rate Coefficients 170
1011 – Eutectic Reaction Temperatures 171
1016 – Zircaloy Post-Breakaway Oxidation Rate Constant Coefficients 171
1017 – Lifetime Parameters for Breakaway Model 171
1018 – Maximum Lifetime for Breakaway Model 172
1020 – Radial Relocation Model Parameters 172
1021 – Channel-Bypass Relocation Time Constant 173
1030 – dT/dz Model Parameters 173
1101 – Fuel-Cladding Gap Emissivities 173
1102 – Steel Emissivity 174

1103 – Particulate Emissivity 174
1104 – Oxidized Zircaloy Emissivity 175
1105 – Graphite Emissivity 176
1131 – Molten Material Holdup Parameters 176
1132 – Core Component Failure Parameters 177
1141 – Core Melt Breakthrough Candling Parameters 178
1151 – Conglomerate Debris Surface Area Coefficients 178
1152 – Surface to Volume Ratio for Fluid 179
1200 – Smoothing of Heat Transfer Coefficients 179
1212 – Laminar Nusselt Numbers 179
1213 – Laminar Developing Flow 180
1214 – Turbulent Forced Convective Flow in Tubes 180
1221 – Laminar Free Convection between Parallel Vertical Surfaces 180
1222 – Turbulent Free Convection between Parallel Vertical Surfaces 181
1231 – Forced Convective Flow over a Spherical Particle 181
1232 – Free Convective Flow over a Spherical Particle 181
1241 – Simplified Nucleate Boiling Curve 182
1242 – Simplified Transition Film Boiling Curve 183
1244 – Debris Dryout Heat Flux Correlation 183
1245 – Downward-Facing Lower Head Heat Transfer Correlations 183
1246 – Heat Transfer Coefficient, Lower Head to Atmosphere 185
1250 – Conduction Enhancement for Molten Components 185
1260 – Quench Model Parameters 185
1270 – Flow Regime Map Parameters 186
1280 – Nusselt-Rayleigh Correlation for Molten Pool 187
1281 – Molten Pool Convection Inertial Time Constant 187
1290 – Molten Pool Convection Model Directional Parameters 188

3.2 Fission and Decay Power Parameters 188

1301 – Chexal-Layman Fission Power/Liquid Level Correlation 188
1311 – Material Fission Power Absorption Efficiencies 189
1312 – Component Fission Power Absorption Parameters 190
1321 – Material Absorption Efficiencies for Decay Heat 191
1322 – Component Decay Power Absorption Parameters 191

3.2.1 Numerical Control Parameters 192

1401 – Timestep Control Parameters 192
1402 – Candling Control Parameter 193
1403 – Point Kinetics Time-Step Control 193
1404 – Temperature Feedback Reactivity Coefficients 193
1405 – Point Kinetics Model 6 Group Parameters 194
1406 – Point Kinetics Nuclear Data 195

3.2.2 Geometric Parameters 195

1501 – Canister Mass/Surface Area Splits 195
1502 – Minimum Component Masses 196
1503 – Core Blockage Parameters 196

1505 – Core Blockage Parameters 196
1504 – Core Cell Volume Consistency Tolerances 197
3.2.3 Lower-Head Mechanical Model Parameters 197
1600 – Model Parameters 197
1601 – Larson-Miller Creep-Rupture Parameters for Vessel Steel 198
1602 – Vessel Steel Elastic Modulus Parameters 198
1603 – Vessel Steel Yield Stress Parameters 199
3.2.4 Support Structure Mechanical Model Parameters 199
1604 – Larson-Miller Creep-Rupture Parameters for Support Structures 199
1605 – Internal Steel Elastic Modulus Parameters 200
1606 – Internal Steel Yield Stress Parameters 200

4 PLOT VARIABLES 200
4.1 Core Mass Variables 201
4.2 Core Flow Area and Component Volume Variables 202
4.3 Core Temperature Variables 202
4.4 Core Energy and Heat Transfer Variables 203
4.5 Core Structural Loading and Damage Variables 205
4.6 Core Oxidation and B_4C Reaction Mass Variables 205
4.7 Core Molten Pool Variables 207
4.8 Core Penetration, Lower Head, Breach and Melt Ejection Variables 209
4.9 Core Accounting Variables 209

5 CONTROL FUNCTION ARGUMENTS 210
5.1 Core Mass Variables 210
5.2 Core Flow Area and Component Volume Variables 212
5.3 Core Temperature and Melt Fraction Variables 213
5.4 Core Energy and Heat Transfer Variables 214
5.5 Core Structural Loading and Damage Variables 216
5.6 Core Oxidation and B_4C Reaction Mass Variables 216
5.7 Core Quench Variables 218
5.8 Core Penetration, Lower Head, Breach and Melt Ejection Variables 218
5.9 Convecting Molten Pool Variables 219
5.10 Core Accounting Variables 220

6 EXAMPLE INPUT 221
6.1 Example MELGEN Input 221

6.2 Example MELCOR Input 225

7 COR PACKAGE OUTPUT 225

8 DIAGNOSTICS AND ERROR MESSAGES 225

APPENDIX 1 229

List of Figures

Figure 1.1 Possible lower head geometries. 20
Figure 1.2 Lower head geometric parameters..... 21
Figure 1.3 Lower head wall thickness. 22
Figure 1.4 Top view of fuel rack layout in the bottom of the spent fuel pool. 28
Figure 1.5 Spent fuel pool rack with assemblies. 29
Figure 1.6 Separate effects spent fuel pool model. 29

1 Introduction

The MELCOR Core (COR) package calculates the thermal response of the core and lower plenum structures, including the portion of the lower head directly beneath the core, and COR models the relocation of core materials during melting, slumping, and debris formation. It also calculates a simplified mechanical response of the lower head to the differential pressure between the lower plenum inside the vessel and the reactor cavity outside the vessel. Multiple structures are modeled as separate components within a single core cell. Intact components include fuel pellets, cladding, canister boxes, pressurized water reactors (PWR) core baffle and formers, and other structures, such as control rods or blades. Canister boxes are only present in boiling water reactors (BWRs), and the core baffle and formers are only present in PWRs. Particulate debris is also modeled as a possible component within a core cell, as are possible molten pools. Each of these components may be composed of several materials (e.g., Zircaloy and ZrO_2 in the cladding), which are maintained in thermal equilibrium. This is discussed in more detail in Section 1.1

All important heat and mass transfer processes are modeled for each cell component. Thermal radiation among the various components within a cell and between cells in both the axial and radial directions is included, as well as radiation to boundary structures (e.g., the core shroud or upper plenum, which are modeled by the Heat Structure [HS] Package) and to a liquid pool. Gap radiation/conduction between fuel and cladding and axial conduction in each of the components is modeled. Convection to the fluid in adjacent control volumes is modeled for a wide range of fluid conditions and structure surface temperatures, including nucleate and transition film boiling. Melting of boundary steel structures, with the addition of the molten steel to the core debris, may be modeled using appropriate input to the HS Package. Oxidation of Zircaloy and steel is modeled for both the limiting cases of solid-state diffusion of oxygen through the oxide layer and gaseous diffusion of steam or oxygen through the gas mixture. The reaction of B_4C with steam is also modeled. Although, the default B_4C oxidation model produces satisfactory results in highly oxidizing atmospheres, it is recommended that the advanced B_4C reaction model be invoked (through input on record COR_B4C) if significant hydrogen concentrations (reducing environment) are expected.

The core degradation model treats eutectic reactions that lead to liquefaction below normal melting points, dissolution reactions that lead to significant fuel relocation below the UO_2 melting temperature, candling of molten core materials (i.e., downward flow and refreezing), and the formation and relocation of particulate debris.

Included in MELCOR 2.1 is the modeling of convecting molten pools, both within the core and in the lower plenum. The core degradation model has been extended to account for separation of the oxide molten pool from the metallic molten pools, partitioning of radionuclides between oxide and metallic phases, uniform distribution of composition and enthalpy throughout a convecting pool, displacement of molten material by solid particulates, heat transfer from a convecting molten pool to the underlying solid substrate, and formation and movement of a localized blockage within a core cell.

Various geometric variables (e.g., cell surface areas and volumes) are updated for changing core geometry. Changes in core flow resistance resulting from relocation of core materials may be modeled, but the connection to the hydrodynamic packages (Control Volume Hydrodynamics [CVH] and Flow Path [FL]) is not automatic. Input on FL_BLK record is required to specify which core cells are associated with each flow path involving the core. Because only CVH and FL model the flow of water and gases, the effects of blockages on circulation can be modeled only to the extent that the CVH/FL nodalization can resolve that circulation.

The remainder of this section briefly describes the nodalization scheme and the calculation framework used in the COR package. More comprehensive information on these topics, as well as on the specific phenomenological models used, is given in the COR Package Reference Manual.

1.1 Nodalization Scheme

The core and lower plenum regions of the reactor vessel are divided into concentric radial rings and axial levels; the numbers of rings and levels are input by the user. A particular radial ring and a particular axial level define a core cell, whose cell number is specified by a pair of two integers; the first integer represents the axial level number, and the second represents the radial ring number. Radial rings are numbered from the center out, and axial levels are numbered from the bottom up.

The scheme described above applies only to structural materials in the core and lower plenum and is independent of the CVH nodalization for the reactor vessel. Each core cell interfaces with a primary (or channel) fluid volume. For BWRs, a secondary, or bypass, fluid volume is specified for cells in the core to represent the interstitial volume between fuel assemblies. For PWRs, a bypass volume may be specified for cells in the outermost ring of the active core to model the region between the core shroud (baffle) and the core support barrel. A single CVH control volume may interface with many core or lower plenum cells.

Each core cell may contain one or more components. A number of distinct intact components are modeled:

(1) Fuel pellets

(2) Cladding

(3) BWR canister walls, split into two parts: one part adjacent and another part that is not adjacent to the control blade (only permitted in a BWR)

(4) PWR baffle (shroud) around the active core (only permitted in a PWR)

(5) PWR core formers between the baffle and the core support barrel (only permitted in a PWR)

(6) Supporting structure

(7) Nonsupporting structure; because of the effects of radiation to the control blade, the two canister parts may have significantly different temperatures.

A core cell may also contain particulate debris (i.e., rubble) resulting from the collapse of fuel rods and other core components. It may also contain molten materials, local to the cell or part of a coherent molten pool that extends through a number of cells. In a BWR, it may reside either inside or outside the channel box, in the channel, or bypass region, respectively. Since version 1.8.6, MELCOR also allows the definition of a bypass region in a PWR, between the core shroud (baffle) and the core support barrel in the outermost ring of the active core.

Particulate debris and molten pools in the channel are distinguished from those in the bypass, using separate components for each. The distinction exists only for core cells that have distinct channel and bypass regions. Even then, most of the distinction is lost when the channel box or core shroud fails, although both sets of materials must continue to be tracked separately because they typically occupy space in different CVH control volumes.

Conglomerate debris (i.e., core material that has melted and resolidified) is modeled as an integral part of the component onto which it has frozen; that component may include any item from the above list except fuel pellets. Heat transfer and oxidation processes are modeled separately for each component.

The following table identifies each component by its component number and component identifier, which are often used in the COR package documentation. The duplication of numbers for CB and SH reflects the fact that, because only one of each pair can occur, the same portion of the database is used for both. Number 5 is assigned to FM component (instead of OS in the previous versions)

1	FU	Intact fuel component
2	CL	Intact cladding component
3	CN	Intact canister component (portion not adjacent to the control blade)
4	CB	Intact canister component (portion adjacent to the control blade)
4	SH	Intact PWR core shroud (baffle)
5	FM	Intact PWR core formers
6	PD	Particulate debris component (in the channel)
7	SS	Supporting-structure component
8	NS	Nonsupporting structure component
9	PB	Particulate debris component in the bypass (if any)
10	MP1	Oxidic molten pool (portion in the channel)
11	MB1	Oxidic molten pool in the bypass (if any)
12	MP2	Metallic molten pool (portion in the channel)
13	MB2	Metallic molten pool in the bypass (if any)

The primary difference between the supporting- and nonsupporting-structure components is the ability to support other core components (core support structures) or not (control rods or blades). Nonsupporting structure component (NS) is intended to represent things like

control rods/blades in PWRs and BWRs, stiffeners in Phebus experiments, and other similar objects that cannot support anything except themselves. All core elements that have the capability of supporting other types of core components are intended to be modeled using the supporting structure (SS) component. These include core plates, and control rod guide tubes in BWRs. Modeling options involving SS and NS are discussed in Section 1.3.

Each component may be composed of one or more materials. Eight materials are modeled in the COR package:

(1) UO_2

(2) Zircaloy

(3) Steel

(4) ZrO_2

(5) Steel oxide

(6) Control rod poison, which may be either B_4C or Ag-In-Cd

(7) Inconel

(8) Electric heating element material (defined by the user).

The last of these is intended for use in analysis of electrically heated experiments. If it is present in a core cell, subroutine ELHEAT will be called to calculate the associated heating power in that cell. MELCOR 2.1 allows specification of electrical heater elements by way of user input.

Power and voltage data are obtained from external data files (EDF) or tabular functions, which must be defined in MELGEN. Resistivity data are obtained from control functions and used with geometric length/area data for the heaters defined or calculated in SUBROUTINE ELHEAT itself. Contact resistance (per rod) is obtained as the value of the control function.

This new input is described in Section 2.9 (MELGEN electric heating user input).

These materials melt, relocate, and freeze separately and independently unless the core materials interactions model has been activated, in which case they form a mixture that melts, relocates, and freezes coherently. Steel and steel oxide are each considered a single material within the COR package, but the user must specify (in Material Properties Package input) the fractions of iron, nickel, and chromium in the steel so that oxidation can be properly treated and the right amounts of each species can be transmitted to the Cavity (CAV) Package during debris ejection. The melting and candling of materials results in the possibility of any or all materials being found in a given component.

A lumped parameter approach is used for each component within a cell; therefore, each component is represented by a single equilibrium temperature. All thermal calculations are done using internal energies of the materials, and the mass and internal energy of each material in each component are tracked separately to conserve total mass and energy to within machine round-off accuracy.

Several geometric variables are used to further describe the cells. For each structural component a surface area is defined for convection and oxidation calculations; the canister and PWR core shroud components have defined surface areas for each side to communicate separately with the channel and bypass control volumes. The effects of conglomerate debris on component surface areas are factored into the oxidation calculations, and for oxidation of debris, separate Zircaloy and steel surface areas are used. Equivalent diameters for each component are also specified for use in various heat transfer correlations. Cell boundary areas for inter-cell radiation (both axially and radially) are defined. Volumes of components and the empty fluid space are tracked for core slumping and flow blockage calculations.

The lower-head model added with MELCOR 1.8.6 is more general than in previous versions. Major changes from MELCOR 1.8.5 are that the bottom of the lower head may be curved and that the model has been extended to include a portion of the lower vessel cylinder (typically, that part that is below the baffle plate in a BWR or the bottom plate in a PWR).

The lower head is modeled as a hemispherical shell, a truncated hemispherical shell, or flat plate of user-defined thickness and composition. The lower-vessel cylinder may have a different thickness, and the transition in thickness may be either at the juncture with the cylinder (as in a typical PWR) or within the head, at the radius of the core (as in a typical BWR).

By default, each will be modeled as a user-defined number of equal thickness meshes of carbon steel through the head or wall. These are numbered, beginning from the outer surface of the head or vessel and progressing inward. The user may also specify the meshing and composition in detail. In addition to stainless steel, which is included in the COR package database, the user may specify the materials in a composite lower head. Hence, with appropriate Materials Properties (MP) Package input the user may model insulation, carbon steel, and a stainless steel liner, if desired. The outer node (surface) communicates with the CVH fluid in the reactor cavity, and downward-facing boiling heat transfer to a flooded cavity is modeled. The inner node (surface) communicates thermally with both the penetrations and the debris.

A further change from previous versions is that there is no longer a one-to-one correspondence between segments of the lower head and rings of the core nodalization. However, no lower-head segment may be interfaced to more than one core cell. For each segment of the lower head the user can define representative penetrations (e.g., instrumentation tubes or guide tubes) in the lower head, specifying their mass and surface area.

Heat transfer from the debris to the lower head and its penetrations is modeled parametrically. A two-dimensional solution is used to determine the temperature profile through the lower head and the lower-vessel cylinder. Based on the Larson-Miller parameter and a life-fraction rule, the lower-head mechanical model can use the temperature profile and the loading of the vessel to predict creep-rupture failure of the lower head.

1.2 Calculation Framework

The COR package uses an explicit numerical scheme for advancing the thermal state of the core through time. To mitigate numerical instabilities, a subcycling capability has been developed to allow the COR package to take multiple timesteps across a single system cycle. All energy generation and heat transfer rates are evaluated at the beginning of a COR package subcycle based on current temperatures, geometric conditions, and an estimate of the local fluid conditions (calculated by the COR package dT/dz model to reflect the temperature variation of a control volume containing many individual core cells). The net energy gain (or loss) across the subcycle is determined for each component by multiplying these rates by the COR package timestep. The temperature change for any component with total mass greater than a user-defined minimum is limited to a user-input maximum; if the calculated temperature change is greater than this limit, the core timestep is reduced accordingly. If the energy input to any fluid volume changes from previous values in such a way as to possibly result in numeric instability between the COR and CVH packages, the system timestep may be immediately cut or a reduction may be requested for the next system cycle.

At the end of the COR package timestep, after the thermal state of the core has been updated, relocation of core materials and debris formation are calculated by the core degradation models. First, molten intact materials are moved from intact components to the conglomerate debris associated with the components. Then, if the materials interactions model has been activated, eutectic reactions are calculated between various intact solid materials within a component if the component temperature exceeds the eutectic points, and dissolution reactions between certain intact solid materials and molten eutectic mixtures are calculated (e.g., intact UO_2 fuel dissolved by Zircaloy-bearing mixtures); these materials are also added to the conglomerate debris. Molten materials are relocated downward by the candling model (provided there is no flow blockage) and radially by the spreading model if there is a significant difference in the liquid levels in adjacent core rings. Intact components are converted to debris if various debris formation criteria are met.

Relocation of particulate debris downwards and radially by gravitational settling is normally modeled at the end of a core subcycle by logical processes through consideration of volume, porosity, and support constraints. Rates are not explicitly considered; however, a limit on the maximum rate is imposed to force the relocation to go to zero for very small timesteps. However, if the user has activated the in-vessel falling debris quench model, then debris relocation immediately following the failure of the core support plate in each radial ring is calculated assuming that the debris falls at a constant velocity. Furthermore, during a short time interval following the failure of the core support plate, heat transfer from the falling debris to the pool of water in the lower plenum is calculated. This model can be

used to simulate energetic fuel-coolant interactions that may occur and threaten the integrity of the reactor pressure vessel.

Whenever mass is relocated or debris formed, energies in the new or changed components are re-evaluated to maintain thermal equilibrium, and any relevant geometric variables are recalculated to reflect the change in geometry. Failure of the core plate (or any other structure modeled as SS and providing support) is predicted based on models described in Section 1.3. Failure of the lower head is assumed to occur based on any number of criteria:

(1) The temperature of a lower head node or penetration reaches a critical temperature defined by the user.

(2) The differential pressure exceeds a user-defined maximum.

(3) The accumulated plastic strain exceeds the failure limit, which may be modified by the user.

(4) The value of a user-defined logical control function is found to be true.

1.3 Support of Core Structures

Originally, MELCOR had a one-size-fits-all 'Other Structure (OS)' component which has since been replaced (MELCOR 1.8.5) in the code by a supporting structure component (such as core support plates) and nonsupporting structure component (such as control rods/blades), SS and NS, respectively. The SS and NS components can represent only one type of structure in each core cell, but they may be used to represent quite different types of structures in different core cells. The OS component was removed from MELCOR 2.0.

NS cannot support anything but itself. It may be supported from below like a BWR control blade, supported from above like a PWR control rod, or fixed in place like elements of certain experimental facilities (e.g., the stiffeners in Phebus). Local failure of NS is based on the remaining thickness of structural metal and on temperature. It will fail if the thickness of metal falls below a critical value or—regardless of the thickness of metal remaining—if the temperature exceeds a limiting value. The structural metal is taken as steel by default, but can be specified to be Zircaloy. The default failure temperature is the melting point of the structural metal. Both the critical thickness and the limiting temperature can be redefined if desired.

SS can support itself and other core components, with several choices available for the models to be applied in each core cell. The choice of models affects the treatment of support and loading, the failure criteria, and the consequences of failure. Structural modeling of SS in a core cell is tied to an input keyword. The built-in options are as follows:

(1) PLATE, representing part of an edge-supported plate that can initially support fuel assemblies and particulate debris above it, and any inner sections of the same plate. On failure, SS modeled as PLATE is converted to particulate debris along

with everything supported by it, including any unfailed inner rings of the same PLATE and anything supported by them.

(2) PLATEG, representing part of a grid-supported or egg-crate plate that can initially support fuel assemblies and particulate debris above it. On failure of SS modeled as PLATEG, only the capability to support particulate debris and intact components in cells above is removed. Thus, everything resting on that ring of the plate will fall, but the plate will remain in place until it melts. The picture corresponds to failure of the plate portion with survival of the grid and closely resembles the modeling using OS in MELCOR 1.8.4.

(3) PLATEB, representing part of a BWR plate that can initially support only particulate debris, although its presence is required to transfer the weight of fuel assemblies and canisters to control rod guide tubes below it. On failure of SS modeled as PLATEB, only the capability to support particulate debris in cells above is removed. Thus, intact fuel and canisters can continue to be supported by the control rod guide tubes below, and the failed plate will remain in place until it melts.

(4) COLUMN, representing the control rod guide tubes in a BWR that can initially support nothing but themselves and, through the mediation of PLATEB, fuel assemblies and canisters above the core plate. On failure, SS modeled as COLUMN is converted to particulate debris, along with anything supported by it including any unfailed upper sections of the same column and any other components whose weight is transferred to it by an element of PLATEB. Since MELCOR 1.8.6, if the lowest element of a column rests on either PLATE or PLATEG, it is considered to be supported by that plate.

(5) ENDCOL, equivalent to COLUMN, except that it is considered to be self-supporting.

SS may be specified as failing when a critical temperature is exceeded or when a specified logical control function becomes true. It may also be specified as failing under load, by yielding, as a consequence of creep, or, where appropriate, by buckling. The stresses are calculated from the total load, based on simple mechanical models appropriate to the form assumed for the SS. More details may be found in the COR Package Reference Manual.

The specification of NS and SS models is very flexible. There are global defaults, which may be based on the core type, that may be overridden by optional user input. The input records involved are COR_NS for NS, and COR_SS for SS. The records are input as tables. If the user redefines the global parameters for individual or several core cells, it should be kept in mind that definitions in later records take precedence over preceding records. Therefore, it is recommended that global definitions for all core cells occur first and that definitions for individual cells occur last.

1.4 User Control of Core Oxidation and Material Relocation

In the oxidation of core components, there is competition among various surfaces for the available oxidizers (steam and/or oxygen). In addition to competition between surfaces within a single core cell, which is treated by internal models described in the COR Package Reference Manual, there is also competition among core cells that interface with the same CVH control volume for a share of the total oxidizer inventory of that control volume.

If one CVH control volume contains (parts of) several rings of the core, the available oxidizers are divided in proportion to the minimum open flow areas in each of those rings within the control volume. (The possibility of cross flows is not considered.) The internal calculation of areas may be used, or the user may choose to specify the flow splits using control functions as specified on the optional COR_RAF record. Minimum fractions may be imposed, using sensitivity coefficient array 1007, to avoid total steam starvation. If several core levels are contained in the ring, each is given access in turn to the total inventory of oxidizers remaining in that ring after oxidation is evaluated for cells that come earlier in the order. Cells are processed in the direction of assumed flow through the ring. By default, this is currently based on the computed direction of flow at the exit plane of the core, but can be overridden by control functions optionally specified on COR_RP record.

Core materials can relocate by melting and candling. They can also fail mechanically and be converted to particulate debris (i.e., rubble), which can then slump to lower parts of the core if there is space available. Materials interactions (formation of eutectics) are frequently involved in these processes. MELCOR has a materials interactions model, activated by input on the COR_MS record, but users of version 1.8.3 reported problems with the model. Although some of these problems were resolved in later versions, we presently recommend that the model be used with great care, if at all. The most important effects of materials interactions can be captured using parametric models.

When materials that form core components melt, they are usually assumed to candle immediately. The exception is molten metal (Zr and stainless steel) held up behind oxide shells; parameters used in the holdup model are part of sensitivity coefficient array 1131. If the materials interactions model is not active, the dissolution of other materials in candling molten metals is modeled by secondary transport parameters input on the COR_CMT record.

Note that molten metal may be associated with an intact component, either held up behind an oxide shell or as conglomerate that is prevented from candling by the absence of free volume in a lower cell.

When the remaining thickness of unoxidized metal in a component in a core cell is reduced below a minimum defined by input on record COR_CCT, the component is converted to particulate debris. Two exceptions are made for fuel rods. First, oxidized fuel rods are assumed not to fail until the temperature has risen to a value that, by default, approximates the eutectic temperature for the UO_2/ZrO_2 system, at which materials interactions would begin to liquefy the solid structure. Second, fuel rods are assumed to fail unconditionally (regardless of other input) if the fuel temperature reaches a value that, by default,

approximates the melting temperature of UO_2. The two threshold temperatures are defined by sensitivity coefficient array 1132.

Intact components are also converted to particulate debris if they lose support. A component in a core cell is supported by existence of the same component in the next lower cell (the next higher cell for NS supported from above). It may also be supported by unfailed SS in the next lower cell.

Particulate debris (PD and/or PB) will slump, subject to constraints of available volume and support by SS. In general, particulate debris cannot fill the entire volume that would be available to fluid in a core cell. The volume available to particulate debris in the channel and/or bypass of any cell is based on the parameters specified for the debris exclusion model in that cell, as input on COR_DX record.

These parameters allow intact components to exclude particulate debris from a volume larger than their physical volume, specified as fractions of the associated total volume, either channel or bypass. (Of course, particulate debris will be excluded from the physical volume of the component in any case.)

The default values will prevent particulate debris from entering the channel region while intact fuel rods are present. In a BWR, they will also prevent it from entering the unbladed portion of the bypass region while there is intact CN, or the bladed portion of the bypass region while there is intact NS representing control blades. See Section 3.2.3 of the COR Package Reference Manual for more discussion of the model.

The porosity of a debris bed specified as PORDP on COR_ZP input record is also considered.

Support is determined by the specification of structural models for SS. SS loses the ability to support particulate debris and other components when it fails. Several models are available for the loading and failure of SS; the choice is defined by input variables on COR_SS record. When SS in a core cell fails, it may or may not be converted to particulate debris, depending on the specific option chosen for SS in that cell.

1.5 Summary of Modeling Changes since 1.8.6

The following subsections present short descriptions of the new and changed modeling approaches and options beginning in MELCOR 1.8.6.

1.5.1 Changes in COR and Lower Head Nodalization

Lower-head modeling in the MELCOR COR package has been substantially modified in MELCOR 1.8.6, and its application has been extended to include the portion of the cylindrical reactor vessel (if any) that is below the bottom of the baffle plate in a BWR or the lower plate in a PWR. This replaces the previous use of the HS package to provide a radial boundary condition for that portion of the core model that is below the bottom of the core barrel. As a result, the input for these lower levels must not include a radial boundary HS on the COR_ZP input record. Instead, a character string NO must be specified for these levels to indicate the absence of such a structure.

The new model also allows representation of a curved lower head. The noncylindrical portion can be a hemisphere with the same radius as the cylindrical portion of the reactor vessel, a lesser portion of a sphere of greater radius, or a flat disk. This requires changes to records COR_GP and COR_RT and the addition of new geometry records, COR_VP and COR_AVP.

Both the region below the curved lower head and that outside the core barrel (essentially the downcomer region in a PWR or BWR) is formally included in the COR nodalization, but is inaccessible to the COR package and is not permitted to contain core materials. The drawing below shows the three possible geometries:

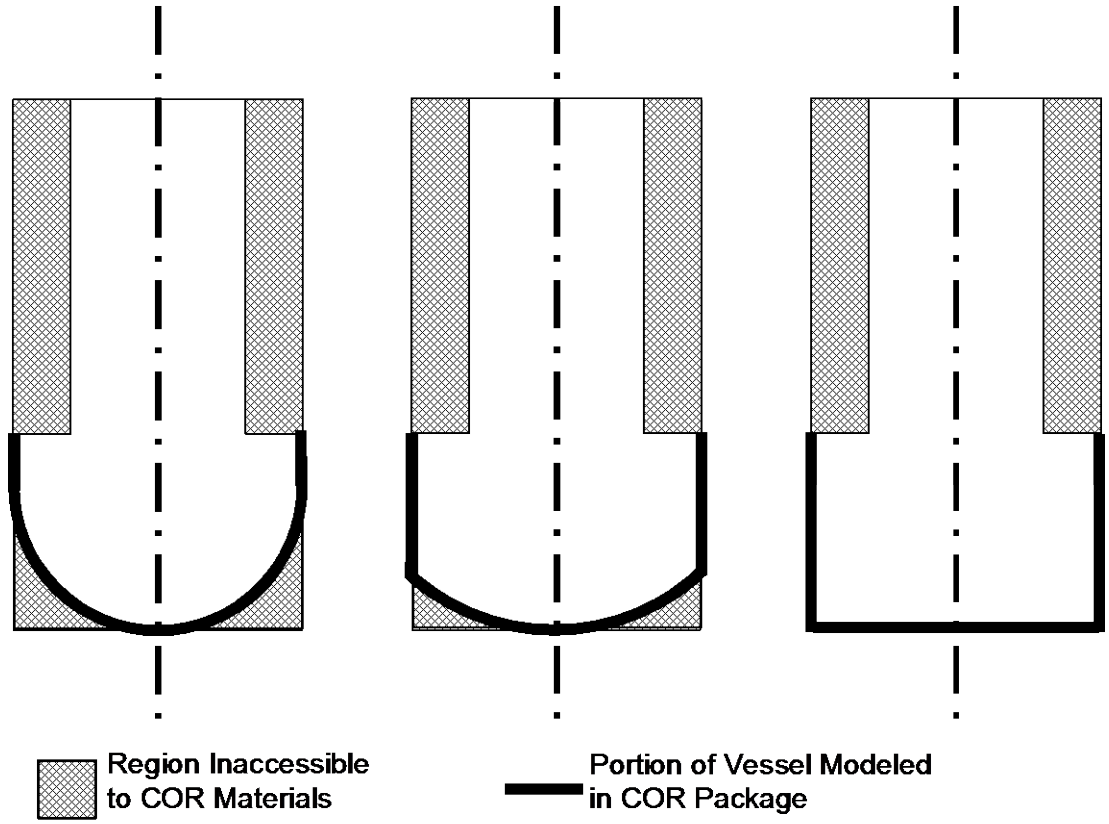


Figure 1.1 Possible lower head geometries.

In addition, while MELCOR 1.8.5 modeled the lower head using only one ring in each ring of the COR nodalization, MELCOR 1.8.6 allows a finer nodalization with one or more head segments in each ring. However, because no segment may be interfaced to more than one core cell, there are restrictions on the definition of the segments in the noncylindrical portion of the head.

The thickness of the lower head may differ from that of the cylindrical vessel, and the transition may take place at either the radius of the core (typical of a BWR) or at the radius of the cylindrical vessel (typical of a PWR).

1.5.2 Core. Lower Vessel, and Lower Head Geometry in MELCOR 1.8.6

The improved representation of the vessel and lower head requires additional geometric information. Required data include the elevation of the BWR baffle plate or PWR bottom plate, HLST, and the elevation of the core support plate, HCSP, which cannot be below HLST. HCSP is used only as the elevation that separates a lower-plenum melt pool from an in-core melt pool. Also required are the inner radius of the vessel cylinder, RVESS, and the outer radius treated in the COR package above HLST, RCOR. For a curved lower head, the radius of curvature of the inside of the lower head, RVLH, is also required and must not be less than RVESS.

Within the COR nodalization there must be ring radii that match RCOR and RVESS, and axial levels that match HLST and HCSP. For a curved lower head, there must also be an axial elevation that matches the intersection of the head and the cylindrical wall.

There may be more than one segment of the lower head in a single radial ring, but no segment may be interfaced to more than one core cell. This requires that there be a segment radius corresponding to each intersection between the noncylindrical portion of the lower head and the boundary of a radial ring or axial level. These dimensions and requirements are shown in the following sketches.

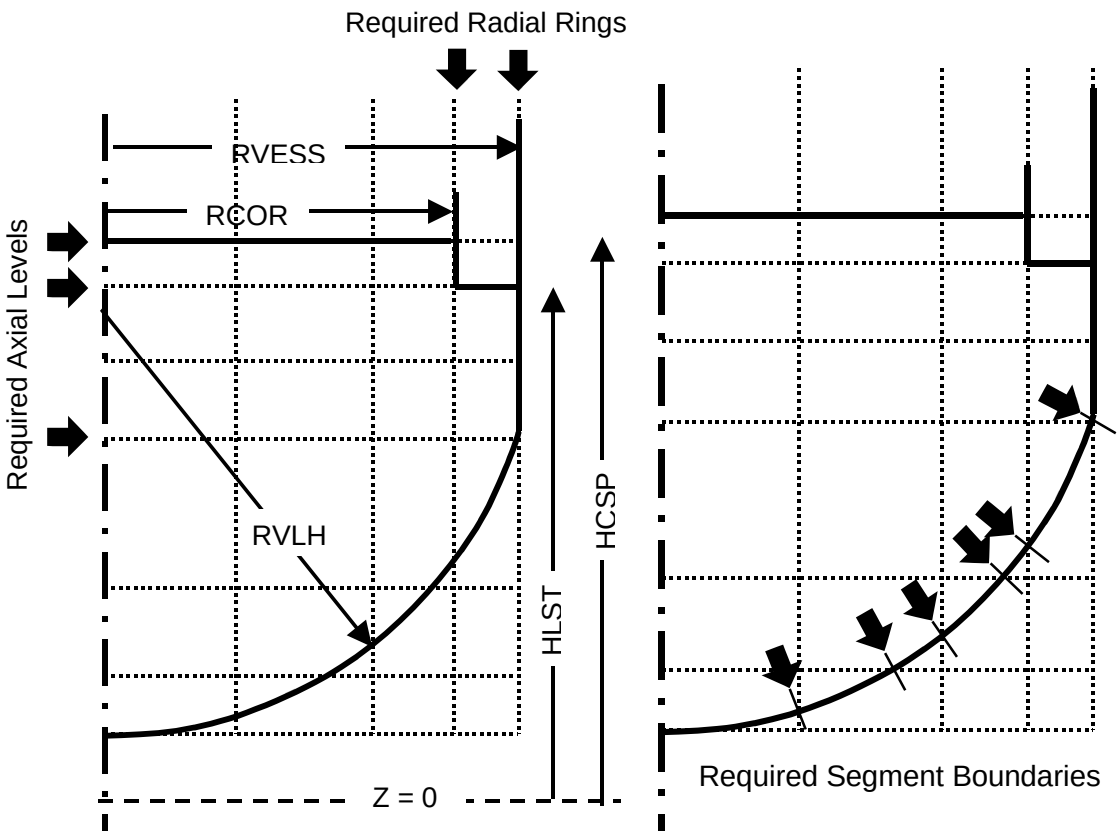


Figure 1.2 Lower head geometric parameters.

Finally, the thickness of the central portion of the lower head, DZLH, may be different from the thickness of the cylindrical vessel wall, DZRV. The same number of temperature nodes, NLH, will be used in each. There are two options for the location of the transition: at the radius of the core barrel, RCOR (typical for a BWR) and at the radius of the vessel, RVESS, (typical of a PWR). These options are illustrated in the following sketches.

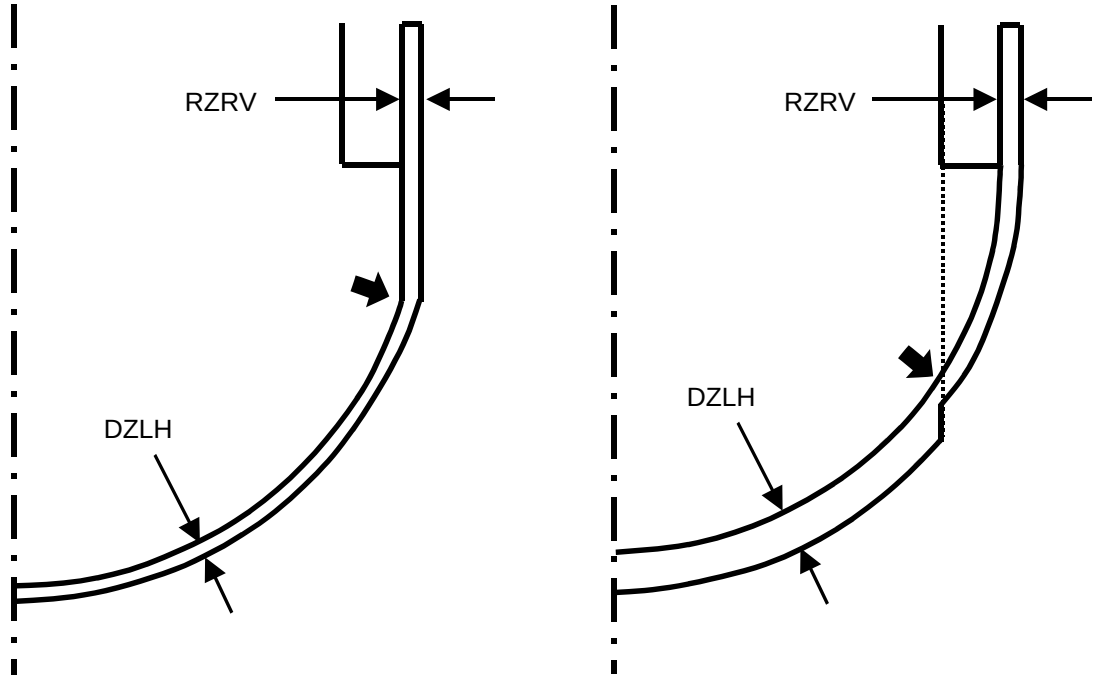


Figure 1.3 Lower head wall thickness.

1.5.3 PWR Core Periphery Model

A typical PWR has an open, unchanneled core. A shroud (i.e., baffle) follows the outer boundary of the fuel assemblies, supported by formers that are attached to the core support barrel. Only a small bypass flow is permitted through this barrel/baffle region so that most of the coolant must pass through the fuel assemblies.

MELCOR 1.8.6 allows this region to be explicitly modeled. The model uses two new components, SH and FM, to represent the shroud (baffle) and formers, respectively. These components are permitted to exist only in the outermost occupied ring of cells above the bottom of the core support barrel. In most cases, the radial-boundary HS in these core levels should represent the core support barrel. A distinct CVH volume may be specified as the bypass volume for such cells and provides fluid boundary conditions for FM, the outside of SH, and the inside of the radial-boundary HS.

There is no provision for the specification of the SH and FM components on cell mass input record COR_CCM; the detailed COR_Kcc input records must be used for any cell containing those components.

1.5.4 Lower Head Model

The major changes from MELCOR 1.8.5 are that the bottom of the head may now be curved, and that there is no longer a one-to-one correspondence between segments of the lower head and the rings of the core nodalization. The user must specify (on COR_LHD input record) the outer radius of the NLHTA lower-head segments that form the bottom (noncylindrical) portion of the head.

Because no lower-head segment may be interfaced to more than one core cell, there must be a segment boundary at each point where the lower head crosses a boundary between radial rings or axial levels. In order to simplify the task of defining consistent input, a new input field RINGR on COR_RP record has been added to define the radii corresponding to ring boundaries. There must be a lower-head segment corresponding to each of these radii.

If the bottom portion of the head is curved (ILHTYP = 0 on record COR_VP), there must also be a lower-head segment corresponding to each intersection of the curved portion of the lower head with axial elevations input as Z on COR_ZP record. Such radii must be calculated from other input, and MELGEN requires that the match be very close (to six figures). However, if approximate values are input, it will compute and report precise values as part of the error message generated.

The model also includes a lower-head segment for each axial level associated with the cylindrical portion of the head, for a total of NLHT≥NLHTA segments. MELGEN expects to read an input record for NLHT segments and will issue a diagnostic message if the number found is incorrect. Finally, MELCOR 1.8.6 allows for the possibility that different segments of the lower head may interface with different volumes in the containment nodalization.

1.5.5 Reflood Quench Model

MELCOR 1.8.6 employs a new model for heat transfer from partially wetted core surfaces that accounts for the possibility of quenching when water recovers heated surfaces. The model is always active in COR package calculations, although sensitivity coefficients may be chosen to disable the quench model. The only user input is in the form of sensitivity coefficients defining the parameters in the quench velocity correlation and in the pool-bridging model. The quench front location and the quenched and unquenched surface temperatures are available as plot variables for the cladding component, CL. These variables are available as control function arguments for all components, and values may, therefore, be plotted for other components by use of EXEC_PLOT input records to add the control function arguments to the plot file.

1.5.6 New Model for Release of Ag-In-Cd Control Poison

Analysis of the FPT1 experiment as ISP46 showed that the creation of aerosols by vaporization of Ag-In-Cd control poison could have a significant effect on overall aerosol behavior. A new model for this release has been added to MELCOR 1.8.6. The model is inactive by default and can be activated only if the control poison is Ag-In-CD. The user has the additional option of whether a release is to be calculated for any molten material or only when that material is candling.

1.5.7 New Model for Release and Oxidation of B_4C Control Poison

The existing B_4C oxidation modeling approach in MELCOR was intended to reflect the oxidation behavior expected for BWR-type control blades. The geometry and form of the B_4C control material affects its subsequent damage progression and oxidation behavior. BWR control blades comprise small stainless steel tubes about 0.25-inch diameter, filled with B_4C powder. The tubes, in turn, are configured in a cruciform shape by an exterior stainless steel sheath. In the original BWR oxidation-damage progression model, it was recognized from experiments that in high-temperature severe accident conditions, a eutectic interaction between the B_4C powder and the stainless steel tubes results in a liquefaction of the tubes at around 1200° C to 1250° C.

This low-temperature liquefaction of the blade material results in relocation of the oxidizing B_4C before high oxidation fractions can be attained. In recognition of this behavior, the oxidation of B_4C exposed to steam at high temperatures was limited to 2 percent in MELCOR. The new model addresses important geometry differences with PWR-type B_4C control elements. In the PWR control element, the B_4C is in a pellet form, encased in a stainless steel cladding. In this geometry the solid B_4C pellet is larger in diameter and more resistant to dissolution by the SS-B_4C eutectic than the smaller powder-filled tubes found in BWRs. This is found to result in greater exposure of B_4C to an oxidizing environment than in the BWR geometry.

The revised B_4C oxidation model is selected via a new record, COR_B4C. The ROD option, together with modified B_4C properties, represents the formation of the SS-B_4C eutectic by replacing the pure B_4C melting point of 2600 K by an effective eutectic liquefaction temperature of 1700 K; this allows the B_4C to be molten when the SS cladding fails, resulting in candling of the control material. Candling will relocate the control poison to lower, presumably cooler, regions of the core, effectively terminating further oxidation. Before this event, the B_4C is allowed to oxidize when temperatures exceed 1500 K, the temperature at which it is assumed that the B_4C pellet is first exposed to steam after the eutectic reaction induces the beginning of liquefaction. In other words, B_4C oxidation is assumed to commence when the temperature exceeds 1500 K, and the material is assumed to fully liquefy and relocate when the temperature exceeds 1700 K. This treatment more mechanistically represents the physical processes affecting B_4C oxidation than the simple fractional oxidation treatment in the previous model.

1.5.8 User-Defined Heat Sources and Heat Transfer Paths

In some calculations, it may be desirable to consider a heat source in a core component that does not correspond to fission, decay, or electrical heating. The capability has been added to allow the user to specify such as the source for one or more components in one or more cells of the core model on COR_QHS input record.

The heat transfer paths modeled within the COR package are appropriate for conventional commercial light water reactors (LWRs). However, paths that are not important in such a reactor may be significant in other applications of MELCOR. The capability has been added to allow the user to define arbitrary additional heat transfer paths between components on COR_HTR input record. The two components need not be in the same core cell. The

modeling may be based on radiation or conduction; in either case, the user is responsible for defining a single input parameter that captures the geometry of the heat transfer path.

1.5.9 Arbitrary Control Function Failure of COR Components

Although some calculations suggest that the existing models may allow fully oxidized fuel rods to stand far too long, no accepted better model is currently available. As an aid to investigating both the consequences of earlier collapse and possible models for it, the capability has been added to allow the user to define a control function to force collapse of fuel rods that would remain standing with the built-in modeling.

1.5.10 Enhancements to Core Support Modeling

Several users of MELCOR 1.8.5 attempted to use the column support structure to model structural elements other than BWR control rod guide tubes. A typical desire was to model a core support where the weight of the core was transmitted to a lower plate through intermediate support columns. This did not work as intended because MELCOR 1.8.5 treats the bottom of a column as self-supporting. Thus, no load is transferred to the lower plate. Further, if the lower plate does fail as a result of debris loading, this failure has no consequences for the columns or any remaining stub fuel assemblies supported by them.

This modeling has been changed. If a plate (PLATE or PLATEG) is present in the level below the bottom of a column, it will receive the load carried by the column, and the column will collapse when the plate fails. (If no such plate is present, the treatment is as in MELCOR 1.8.5.)

As part of the change, a distinct SS type name, ENDCOL, is now assigned to those elements of COLUMN that are self-supporting. The change is performed automatically as part of input processing for those elements that are self-supporting under the interpretation above. The name is also accepted as input on COR_SS input record and can be used to define self-support for interior elements of what would otherwise be considered to be a continuous column.

In addition, there is a new, limited capability for the user to define additional types of SS with properties specified by input. One possible use is as a grid spacer that can support only particulate debris and fails based on a user-defined parametric condition, such as temperature in excess of a failure value or failure of the associated fuel rods.

In the interest of greater flexibility in failure calculations, the load carried by support structure has been made available as a control function argument.

A pressure differential feature was added to allow loading of SS for stress calculation for the Larson-miller creep rupture model and allow ORing of the stress calculation to a logical control function for failure.

1.5.11 Added Flexibility in Flow Resistance (Blockage) Model

Flow resistance in a debris bed is based on the Ergun equation. The porosity used in the equation is calculated as the ratio of fluid volume (i.e., volume not occupied by solid or molten core materials) to total volume. In some cases, release of fission products may

result in swelling of fuel and debris, which can reduce the connected porosity. This will increase the flow resistance of a debris bed compared to the model in MELCOR 1.8.5.

In order to allow this effect to be captured in calculations, an option was added to allow reduction of the value of porosity used by the Ergun equation by a user-defined factor, specified by a control function. The reduction factor can be defined separately for channel and bypass regions.

1.5.12 Enhanced FU Component Composition

The intact fuel (FU) component normally models the UO_2 pellets in normal LWR fuel, but can also contain a heating element material to allow modeling of electrically heated experiments. The presence of the heating element material anywhere in the core disables models for collapse and relocation of fuel rods. MELCOR 1.8.6 allows for the inclusion of one or two additional materials in the FU component. Collapse models remain disabled if the heating element material is present.

One additional material from any of the other six materials recognized by the COR package (Zircaloy, ZrO_2, stainless steel, stainless steel oxide [SSOX], control poison, and Inconel) can be specified as part of MELGEN input. Control poison can be Ag-In-Cd or B_4C, depending on input on the COR_RT record; the default is B_4C. Although Inconel is tracked separately, it uses stainless steel properties and does not oxidize. Note that COR_MAT input record can be used to substitute the properties of other materials for the default UO_2, Zircaloy, ZrO_2, stainless steel, and/or SS Oxide.

If the user-defined material is Zircaloy or stainless steel, the corresponding oxide will also be allowed as a material in the FU component. Otherwise, no second added material is allowed.

As coded in MELCOR 1.8.6, materials in the FU component will neither oxidize nor melt and candle while they remain part of that component. These processes can only occur after fuel rods have failed and the FU component has been converted to particulate debris.

1.5.13 Enhanced Flexibility in Composition of Other Components

The user may now specify the material in the cladding (CL), canister (CN and CB), and PWR core shroud (SH) components directly, using input on COR_CLM, COR_CNM, and COR_SHM records. The material must be one of those recognized by COR. If it is Zircaloy or steel, oxidation is properly modeled (not the case if substituted material properties are defined using COR_MAT input), and the oxide is also included in the composition.

1.5.14 Improved Local Temperature (dT/dz) Model

A modification was made to the dT/dz model to improve coupling between calculations in COR and those in CVH under conditions of little or no flow. Under near-stagnant conditions, the two calculations could diverge dramatically; if the CVH temperature is driven below 273.15 K or above 10000 K, the calculation will stop. The model (which is used only by COR) was modified to operate on the assumption that there is a characteristic time for recirculation of fluid within each CVH volume, independent of flows through the volume. This appears to be sufficient to stabilize the calculation.

In addition, a change was made to the relaxation (smoothing) of flows. The previous algorithm was found to cause problems when the time scale for flow changes was much less than the relaxation time defined by sensitivity coefficient TRLXZ = C1030(2). (The problem case involved a sudden onset of steaming following inflow of liquid water to a hot region of the core.) The old and new flows are given relative weights of exp(-dt/TRLXZ) and[1–exp(-dt/TRLXZ)], respectively. The algorithm was modified to limit the relative weight of the old flow to a value given by a new sensitivity coefficient array element. (The change in the default value cited for TRLXZ reflects an error in the MELCOR 1.8.5 manual; the code was not changed.)

1.5.15 Access to Emissivity Correlations via Sensitivity Coefficients

The emissivities of core materials were hard-coded in MELCOR 1.8.5. In MELCOR 1.8.6, the correlations have been recorded using sensitivity coefficients to allow some flexibility to modify them from user input. The default correlations are unchanged with one exception: extrapolation of the original correlation for oxidized Zircaloy to a very large oxide thickness could return a negative value; in MELCOR 1.8.6, the correlation cuts off (by default) at an oxide thickness of 1 mm.

1.5.16 Direct Separate Input for CN and CB Geometry

MELCOR 1.8.6 allows separate, direct input of the areas and hydraulic diameters of the channel and bypass sides of canister components CN and CB.

1.5.17 Fuel Rod Collapse Model Logic

A framework for a fuel collapse model with sensitivity coefficients for residual lifetime failure criterion (i.e., time at temperature) has been added. The user may now choose to supplant the default temperature-based failure criterion when the cladding thickness falls below the minimum specified on record COR_ROD. In order to activate this model, the user must also specify time-at-temperature damage function in the form of a tabular function on record COR_ROD.

1.6 Specification of Thermal Conductivity of Gap Gases

MELCOR 1.8.6 permits use of any gas defined in the Non-Condensible Gas (NCG) package as gap material. The data input for this gas on record NCG_PRP will be used as default. In some specialized applications, it may be desirable to model thermal conduction in regions where the effective conductivity is not simply a function of temperature. (The motivating example involved a heat structure containing a gas gap that closed at high temperatures as a result of differential thermal expansion.) If the user chooses to override or modify the default data, tabular or control function may be specified on the last optional fields of Record COR_BCP to calculate thermal conductivity.

1.7 Use of COR Package for Special Fuel Configurations

The core package has been extended to allow users to model other fuel configurations. However, the user is warned that use of these features requires extensive knowledge of how the COR package works. In this section, these features are explained, within the context of the tested fuel configurations, to guide the user through some of the difficulties.

1.7.1 Modeling Accident Scenarios in Spent Fuel Pools

Several code enhancements of the COR package were made to MELCOR to facilitate the analysis of spent fuel pools (SFP) accident scenarios. The COR package already includes fuel degradation models for BWR and PWR fuel, radiation, convection, and conduction heat transfer models, air and steam oxidation models, hydrogen burn models, two-phase thermal-hydraulic models, and fission product release and transport models. Hence, it already contains the basic models to address questions and phenomena expected during a spent fuel pool accident (see the COR Reference Manual for additional details).

Starting with 1.8.6, MELCOR added two new features to the COR package that are specific to SFP modeling: (1) a new rack component, which permits modeling of a SFP racks, and (2) an enhanced air oxidation kinetics model. The new spent fuel pool rack component permits separate modeling of the SFP rack and radiative heat transfer between the rack and existing COR components. The new air oxidation kinetics model predicts the transition to breakaway oxidation kinetics in air environments on a node-by-node basis. These new features can be used to analyze a wide range of postulated SFP accidents. These new SFP features can be used to perform three types of SFP calculations: (1) normal operating conditions, (2) partial loss-of-coolant inventory accident, and (3) a complete loss-of-coolant inventory accident.

Normally SFPs are constructed of reinforced concrete and include a stainless steel wall and floor lining. The high density SFP racks provide spent fuel storage at the bottom of the fuel pool, and are covered with water for radiation shielding under normal SFP operating conditions (see Figure 1.4). The pool walls are modeled as heat structures (HS) and the pool water is modeled using the CVH package.

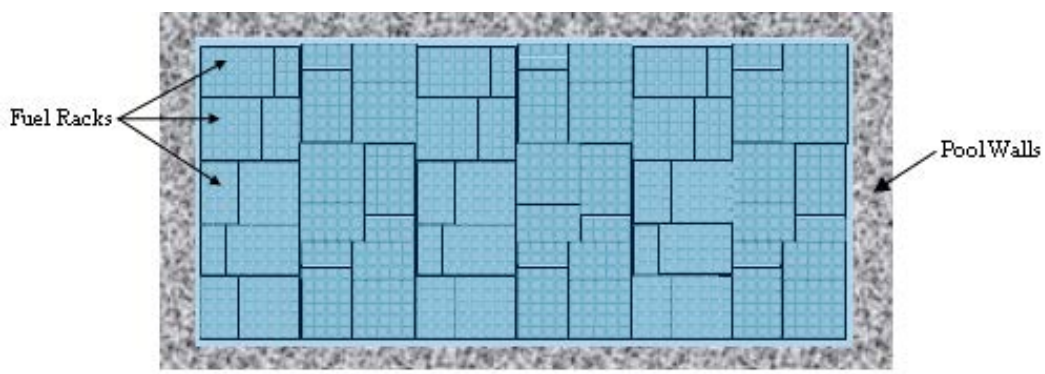


Figure 1.4 Top view of fuel rack layout in the bottom of the spent fuel pool.

The racks are rectilinear in shape and are arranged on the floor of the SFP. The rack module is composed of fuel cell assemblies, a base plate, and base support assemblies (see Figure 1.5). The base plate has chamfered through holes centered at each storage location, which provides a seating surface for the fuel assemblies. These holes also provide passage for coolant flow. Each fuel cell in the rack is composed of a full-length enclosure constructed of stainless steel and often has neutron absorbing material in the side panels for reactivity control. The racks have support assemblies that elevate them above the

bottom pool. Below the rack base plate is the lower gap region which allows water to freely flow under the racks and circulate through the fuel assemblies.

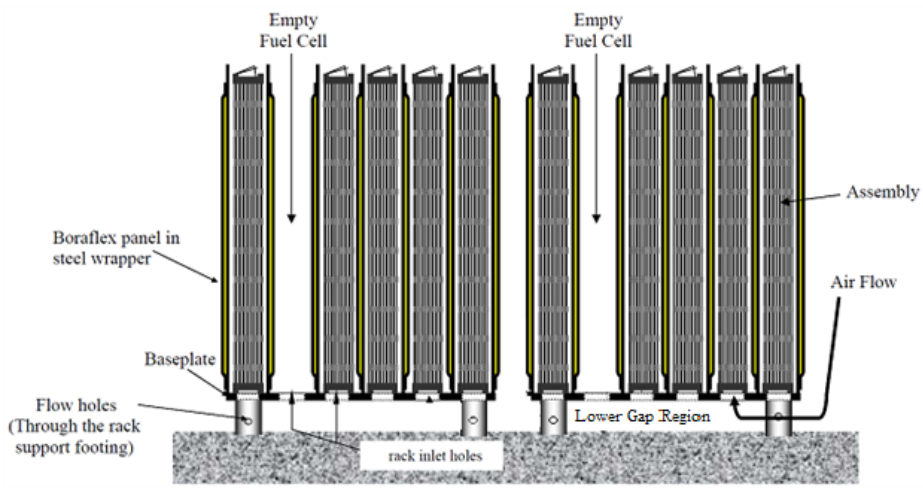


Figure 1.5 Spent fuel pool rack with assemblies.

The SFP racks and the lower gap region below the SFP racks can be modeled using the existing core and lower plenum components. The MELCOR COR package is designed in two-dimensional cylindrical geometry and nodalization of the SFP must fit within this framework. Implicit in this framework is the assumed direction of heat and mass transfer between adjacent rings and adjacent elevations. For SFP models, the user can take advantage of this pre-existing framework and arrange the fuel-rack cells in a similar ring pattern. For example, Figure 1.6 shows a separate effect spent fuel pool successfully modeled with MELCOR using the cylindrical ring patterns on the right and by carefully specifying "ring" volumes and areas.

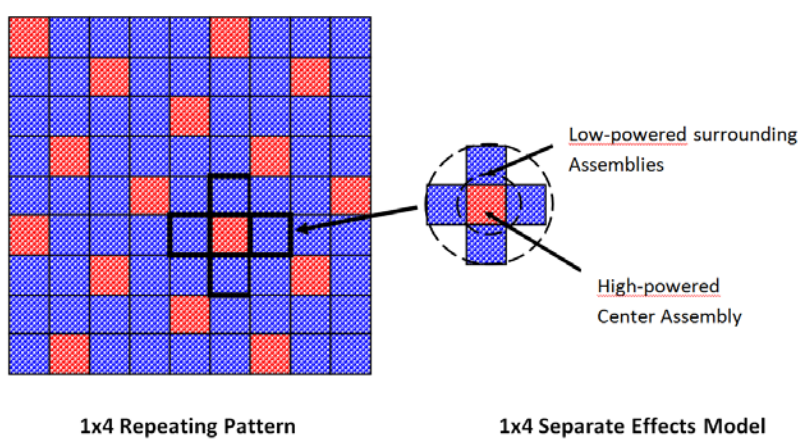


Figure 1.6 Separate effects spent fuel pool model.

Detailed analysis of more complex spent fuel pool networks can be performed if the user is careful and cognizant of MELCOR's limitations. Spent fuel pools may contain thousands of assemblies that span decades of offloads from the reactor. It is possible to specify the average decay heat for each block (ring) of fuel assemblies but it will be done by scaling specific radionuclide decay heat from a single reference time. Similar assemblies can then be grouped together into blocks with generalized thermal communication between adjacent blocks by specifying component-to-component heat transfer paths (i.e., COR_HTR). However, spreading between blocks will still be calculated based on the cylindrical geometry framework.

Finally, there is no comparable lower head component in the spent fuel pool. Since the COR package is hardwired to include the lower head and its relevant physics, the user needs to specify an equivalent lower head component. One approach is to let the lower head component represent the pool floor and assign it appropriate material properties. In this way the lower head should be modeled as a flat plate of user-specified composition and should have a uniform thickness even at the juncture with the edge of the rack. The COR package is hardwired to include a CVH volume below the lower head to represent the cavity volume below the core vessel. One approach is to define this volume as a constant temperature and pressure boundary volume.

The COR package limitations can be summarized as follows:

(1) The SFP nodalization must be adapted from the MELCOR 2-D cylindrical coordinate framework. This may require specification of additional conduction paths between blocks (rings)

(2) There are no analogous mass transfer paths between blocks. In most cases, downward relocation of relocating mass dominates until material is retained in the lower head.

(3) The lower head structure in the COR package cannot be disabled in MELCOR 1.8.6. This requires thoughtful consideration by the user in specifying vessel properties, failure parameters, and possibly spreading parameters.

2 Input Requirements

This section gives the input requirements for the MELCOR COR package, including a short description of the input quantities, their units, and default values, if any. Further description of the input variables and their meaning in the models can be found in the COR Package Reference Manual. It is strongly recommended that the beginning user consult both documents (as well as other package users' guides) when first learning these input requirements.

The first input record for the COR package must be COR_INPUT; then the other records follow. Input record identifiers for the COR package all begin with the character string COR; several are required, but most are optional with defaults provided or obtained from previous

input for other cells. Records from Section 2.1 are used to specify general information about the geometry, nodalization, and physics modeling for the core and lower plenum as a whole. Record COR_ZP is used to specify information about axial levels. Record COR_RP is used to specify information about radial rings. Records from Section 2.4 are used to specify information about individual core cells. Records of the form COR_Kcc and COR_CDM record may optionally be used to define the initial masses of core components in a flexible form that allows specification of a degraded state, including oxidation, particulate debris, and conglomerate debris on all possible components.

Records COR_LHD, COR_LHN, and COR_PEN are used to specify information about the lower head and associated penetrations. Although their use is no longer either required or recommended, record COR_TIN may optionally be used to specify the manner in which the inlet temperature for a core control volume is calculated in the core dT/dz model.

H_2 and CO must be defined in the input for the NCG package whenever the COR package is active. For BWR cores containing B_4C, the gases O_2, CO_2, and CH_4 must also be defined in NCG input.

2.1 MELGEN General Core

COR_DFT – COR Package Default Scheme Record

Optional.

This record modifies a set of default and sensitivity coefficient values in the COR package to those 1.86 standards. Note that these values may not reflect current MELCOR best practice modeling parameters and sensitivity coefficients for the analysis of severe accidents. If the record is present, one floating point field must be present.

(1) DEFAULT

Mode of default scheme. Permitted values are 1.86 and 2.0.

(type = real, default = 2.0, units = none)

If the DEFAULT field is 2.0, no change is made to the existing default values. If the DEFAULT field is 1.86, the following COR package default and sensitivity coefficient values will be modified, unless otherwise specified by the user input record.

Record	Field	DEFAULT = 1.86	DEFAULT = 2.0
COR_RF	FCELR	0.25	0.1
	FCELA	0.25	0.1
COR_CHT (W/m^2-K)	HFRZUO	1000.0	7500.0
	HFRZZR	1000.0	7500.0
	HFRZSS	1000.0	2500.0
	HFRZZX	1000.0	7500.0
	HFRZSX	1000.0	2500.0
	HFRZCP	1000.0	2500.0

Sensitivity Coefficient	DEFAULT = 1.86	DEFAULT = 2.0
1001 (1, 2)	50.4 $Kg^2(Zr)/m^4$-s	26.7 $Kg^2(Zr)/m^4$-s
1001 (2, 2)	14630.0 K	17490.0 K
1001 (3, 2)	0.0 $Kg^2(Zr)/m^4$-s	26.7 $Kg^2(Zr)/m^4$-s
1001 (4, 2)	0.0 K	17490.0 K
1001 (5, 2)	10000.0 K	9998.0 K
1001 (6, 2)	10000.0 K	9999.0 K
1004 (1)	1100.0 K	1100.0 K
1250 (1)	3200.0 K	2800.0 K
1505 (1)	0.001	0.05
1505 (2)	0.001	0.05
1600 (1)	0.0	1.0
1603 (2)	1800.0 K	1700.0 K

Example

```
!        DEFAULT
COR_DFT  1.86
```

COR_GP – Geometric Parameters

Optional.

This record specifies key geometric information for the fuel rods, control rods, and (for BWRs) canister boxes. If this record is present, it may contain from zero to four fields. Default values will be retained for those fields that are not present and for any that contain a negative value. Different default values are defined for a BWR (including and SFP-BWR) or a PWR (including SFP-PWR) or an ACR-700, based on IRTYP input on record COR_RT. These are marked as (B) and (P) and (A), respectively, in the list below. Note that DZLH has been moved to the COR_VP record.

(1) RFUEL

Outer radius of the fuel pellets in the fuel rods.

(type = real, default = 0.00521 (B) or 0.00464 (P) or 0.00629 (A), units = m)

(2) RCLAD

Outer radius of the fuel rod cladding.

(type = real, default = 0.00613 (B) or 0.00536 (P) or 0.00613 (A), units = m)

(3) DRGAP

Thickness of the gas gap between fuel pellets and cladding.

(type = real, default = 0.00012 (B) or 0.00011 (P) or 0.00004 (A), units = m)

(4) PITCH

Center-to-center spacing of the fuel rods.

(type = real, default = 0.0162 (B) or 0.0143 (P) or 0.0162 (A), units = m)

Example

```
!       RFUEL         RCLAD          DRGAP         PITCH
COR_GP  0.3785-02     0.455E-02      0.15E-03      0.1275E-01
```

New core types recognized are SFP-PWR and SFP-BWR.

COR_RT – Reactor Type

Optional.

This record specifies the reactor type, the control rod poison material, and the electric heating element material. This record is not required, but if included, at least the first field must be present. The specification of electrode materials is only permitted for cases involving electrical heating.

(1) IRTYP

Reactor type. The choices allowed are:

(a) BWR
(b) PWR
(c) SFP-BWR
(d) SFP-PWR
(e) ACR-700
(f) PBR
(g) PMR

(type = integer / character*7, default = BWR)

(2) MCRP

Control rod poison type. Only choices allowed are: B_4C, B4C-INT, and AG-IN-CD.

(type = character*8, default = B_4C)

(3) MATHT

Electric heating element material.

(type = character*24, default = URANIUM-DIOXIDE)

(4) MATEL1

First electrode material for electrical heating.

(type = character*24, default = URANIUM-DIOXIDE)

(5) MATEL2

Second electrod material for electrical heating.

(type = character*24, default = URANIUM-DIOXIDE)

Example

```
!       IRTYP  MCRP     MATHT
COR_RT PWR    B4C      URANIUM-DIOXIDE
```

COR_VP – Vessel Parameters

Required.

This record specifies key geometric information for the vessel, including the core region. Different default values are defined for a BWR, a PWR, a SFP-BWR, a SFP-PWR, or an ACR-700, based on IRTYP input on record COR_RT. If a negative real number is input for any real variable with a default value, then default value will be used. All fields are optional.

(1) RCOR

Outer radius for the COR package in active core region.

(type = real, default = see table below, units = m)

(2) RVESS

Radius of the inside of the vessel cylinder. If RVESS = RVLH, the lower head is a full hemisphere; if RVLH>RVESS, it is a truncated segment of a hemisphere.

(type = real, default = see table below, units = m)

(3) ILHTRN

Reactor lower head transition type:

(a) 0 or RCOR
Transition is at RCOR, as in typical BWR.

(b) 1 or RVESS
Transition is at RVESS, as in typical PWR.

(type = integer / character*5, default = 1)

(4) DZRV

Thickness of cylindrical vessel wall.

(type = real, default = see table below, units = m)

(5) DZLH

Thickness of the lower head inside the transition radius specified by ILHTRN on this record.

(type = real, default = see table below, units = m)

(6) ILHTYP

Reactor lower-head type:

(a) 0 or HEMISPHERE
Lower head is a (portion of a) hemisphere.

(b) 1 or FLAT
Lower head is flat.

(type = integer / character*10, default = 0)

(7) RVLH

It is used only for ILHTYP = HEMISPHERE or ILHTYP = 0. Radius of curvature of inside of lower head. For a curved head, it cannot be less than RVESS, which is entered above.

(type = real, default = see table below, units = m)

RCOR and RVLH must each match one of the ring radii input as RINGR on COR_RP record

Table for default values:

Reactor Type	PWR	BWR	SFP-BWR	SFP-PWR	ACR-700
RCOR	1.8796	2.376	2.376	1.8796	2.376
RVESS	2.2392	3.34	3.34	2.2392	3.34
DZRV	0.2144	0.17779	0.17779	0.2144	0.17779
DZLH	0.13	0.22	0.22	0.13	0.22
RVLH	2.2392	3.34	3.34	2.2392	3.34

Example

```
!      RCOR    RVESS   ILHTRN  DZRV    DZLH    ILHTYP     RVLH
COR_VP 1.8E+00 2.2E+00 RVESS   2.2E-01 2.2E-01 HEMISPHERE 2.3E+00
```

COR_AVP – Additional Vessel Parameters

Required.

This record specifies additional geometric information for the vessel. Two fields are required.

(1) HLST

Elevation of the BWR baffle plate or PWR bottom plate. The portion of the vessel that is below HLST and is treated by the lower-head model.

(type = real, default = none, units = m)

(2) HCSP

Elevation of the core support plate, which cannot be below HLST. HCSP is used as the elevation that separates a lower-plenum melt pool from an in-core melt pool.

(type = real, default = none, units = m)

HLST and HCSP are absolute elevations (relative to the universal MELCOR reference); they are not relative to the bottom of the vessel. Each must match one of the axial elevations input as Z on the COR_ZP record.

Example

```
!         HLST        HCSP
COR_AVP   1.97845     2.26471
```

COR_RF – Radiative Exchange Factors

Optional.

This record specifies the radiative exchange factors used to model thermal radiation in the core. These exchange factors roughly correspond to the traditional view factors describing the geometric orientation between two pairs of surfaces. This record is not required, but, if included, from one to five fields must be present. Each value must be a nonnegative real number less than or equal to 1.0. For more details on the interpretation given to these parameters, see the COR Package Reference Manual, Section 2.1.2.

(1) FCNCL

Radiative exchange factor for radiation from the canister wall to the fuel rod cladding. A value must be entered for PWRs, but is not used.

(type = real, default = 0.25, units = none)

(2) FSSCN

Radiative exchange factor for radiation from NS (e.g., control blades) to the adjacent canister walls or to fuel rods and debris if canister is not present.

(type = real, default = 0.25, units = none)

(3) FCELR

Radiative exchange factor for radiation radially outward from the cell boundary to the next adjacent cell. Alternatively, the word 'MODEL' can be input and MELCOR will calculate the value based on an internal model.

(type = real or character*7, default = 0.1, units = none)

(4) FCELA

Radiative exchange factor for radiation axially upward from the cell boundary to the next adjacent cell.

(type = real, default = 0.1, units = none)

(5) FLPUP

Radiative exchange factor for radiation from the liquid pool to the core components.

(type = real, default = 0.25, units = none)

Example

```
!         FCNCL   FSSCN   FCELR   FCELA   FLPUP
COR_RF    0.25    0.25    0.25    0.25    0.25
```

COR_TP – TP Interface and Fission Power and Gap Conductance Control Functions

Required.

This record specifies the Transfer Process (TP) package translation matrix name (for transfer of mass and energy to the CAV package); it may also specify the fission power and fuel-cladding gap conductance control function names. This record is required and must contain at least the first field.

(1) NTPCOR

In transfer process name (information for this *in* transfer process must be input on the TP_IN record) that specifies the input for transferring masses and energies from the COR package to the CAV or FDI packages. In this case name of package NAMEPAC for this *in* transfer process on the TP_IN record must be COR. The number of masses and the number of thermodynamic variables is automatically defined as NMSIN=6 and NTHRM=9. Also, on the corresponding TP_MTX record, NCOL=6. The order of masses ejected for generating the translation matrix is (1) UO_2, (2) Zr, (3) steel, (4) ZrO_2, (5) steel oxide, and (6) control poison. Inconel is included with steel automatically.

For convenience, a character string NO, indicating no TP, may be used for calculations that will not fail the lower head and eject debris. See the TP Package Users' Guide for more details.

(type = character*16, default = none, units = none)

(2) RNTPCOR

In transfer process name for associated radionuclides (in addition to the one [NTPCOR above] for debris).

This field is required if the RadioNuclide (RN) package is active; otherwise, a character string NO must be input. See the TP Package Users' Guide for more details.

(type = character*16, default = none, units = none)

(3) ICFGAP

Fuel-cladding gap conductance control function name. If NO or omitted, no additional gap resistance is calculated. If a control function name is entered, the conductance is added serially to the gap gas conduction for all cells with fuel rods, in parallel to the gap radiation.

(type = character*16, default = NO, units = none)

(4) ICFFIS

Fission power control function option. If NO or omitted, no fission power is calculated.

(a) 0 or NO
No fission power is calculated.

(b) 1 or LIQLEVALL
Control function CFNAME (fifth field on this record) is used to calculate the liquid water level to be used in the Chexal-Layman fission power/liquid level correlation (see Section 2.6 of the COR Package Reference Manual).

Fission power is distributed over all core cells (not lower plenum cells) on a per fuel mass basis.

(c) -1 or FISPOWALL
Control function CFNAME (fifth field on this record) is used to calculate the whole-core fission power. Fission power is distributed over all core cells (not lower plenum cells) on a per fuel mass basis.

(d) 100 or LIQLEVLIQ
Control function CFNAME (fifth field on this record) is used to calculate the liquid water level to be used in the Chexal-Layman fission power/liquid level correlation (see Section 2.6 of the COR package Reference Manual). Fission power is distributed only over liquid-covered cells containing intact fuel.

(e) -100 or FISPOWLIQ
Control function CFNAME (fifth field on this record) is used to calculate the whole-core fission power. Fission power is distributed only over liquid-covered cells containing intact fuel

(type = integer / character*16, default = 0, units = none)

(5) CFNAME

Name of the fission power control function.

This field is required if ICFFIS not equals 0 or NO.

(type = character*16, default = none, units = none)

Examples

```
!       NTPCOR RNTPCOR ICFGAP ICFFIS      CFNAME
COR_TP  TPin01 Tpin501 NO     NO
COR_TP  TPIN02 RNTP5   CF2    LIQLEVALL   CF2
COR_TP  TPIN_1 NO      NO     -1          CntrFun5
```

COR_CHT – Candling Heat Transfer Coefficients

Optional.

This record specifies the refreezing heat transfer coefficients to be used in the candling model for each of the molten core materials. This record is not required, but if included, from one to six fields must be present. Each value must be positive. Due to a large degree of phenomenological uncertainty, it is very difficult to justify particular values for these coefficients. The default values are order-of-magnitude estimates that appear to produce plausible simulations of relocation phenomena, but they should be varied in sensitivity studies to determine their impact on overall melt progression behavior. For more information on how these quantities are used in the candling model, see the COR Package Reference Manual, Section 3.1.

(1) HFRZZR

Refreezing heat transfer coefficient for Zircaloy.

(type = real, default = 7500.0, units = W/m^2-K)

(2) HFRZZX

Refreezing heat transfer coefficient for ZrO_2.

(type = real, default = 7500.0, units = W/m^2-K)

(3) HFRZUO

Refreezing heat transfer coefficient for UO_2.

(type = real, default = 7500.0, units = W/m^2-K)

(4) HFRZSS

Refreezing heat transfer coefficient for steel.

(type = real, default = 2500.0, units = W/m^2-K)

(5) HFRZSX

Refreezing heat transfer coefficient for steel oxide.

(type = real, default = 2500.0, units = W/m^2-K)

(6) HFRZCP

Refreezing heat transfer coefficient for the control poison material.

(type = real, default = 2500.0, units = W/m^2-K)

Example

```
!         HFRZZR   HFRZZX   HFRZUO   HFRZSS   HFRZSX   HFRZCP
COR_CHT   1000.0   1000.0   1000.0   1000.0   1000.0   1000.0
```

COR_MS – Model Switches

Optional.

This record activates or deactivates various models and options. The default values of IHSDT, IDTDZ, and ICORCV are strongly recommended.

Problems with the materials interaction model in MELCOR 1.8.3 have been reported. Some of these have been resolved in later versions, but use of the model is not recommended at this time. As discussed in Section 1.4, many of the effects of materials interactions may be captured without activating the model.

The core model in MELCOR 1.8.6 allows representation of supporting structures (such as core support plates) and nonsupporting structures (such as control rods/blades) as separate components, SS and NS, respectively. The single "other structure" component, OS is not retained in the F95 version.

The core model in MELCOR 1.8.6 allows a consistent representation of channel and bypass volumes in the COR package and that in CVH to be maintained for all reactors as core degradation progresses. This was not true for BWRs in versions prior to 1.8.5. The initial representations are now compared as part of input processing. Apparent discrepancies are noted, and comparison tables are provided as part of MELGEN output. Warning and/or error messages may also be generated.

If a severe discrepancy is found, no restart file will be written. The specific requirement is that the initial total fluid volume in COR lying within any volume/altitude (V/A) segment of a CVH control volume cannot exceed the fluid volume in that V/A segment. If this condition is met, filling the entire fluid volume in COR with debris cannot overfill CVH. The requirement may be bypassed by setting the model switch ICORCV=1.

(1) IEUMOD

Materials interactions model switch.

(a) 0 or INACTIVE
Model is inactive

(b) 1 or ACTIVE
Model is active.

(type = integer/character*8, default = 0, units = none)

(2) IHSDT

HS boundary condition option switch.

(a) 0 or REQ

dT/dz boundary condition option required for core radial boundary structures input on record COR_ZP.

(b) 1 or NOTREQ
dT/dz boundary condition option not required.

(type = integer/character*6, default = 0, units = none)

(3) IDTDZ

dT/dz inlet specification option switch.

(a) 0 or CVH
dT/dz inlet flow and temperature from default hydrodynamic calculation in CVH/FL. For this option, the COR_TIN record is not allowed as part of MELGEN input.

(b) 1 or MELCOR1-8-3
dT/dz inlet flow and temperature calculated as in MELCOR 1.8.3, with the addition of a downward-flow option. For this option, the COR_TIN record is required for at least the lowest and highest hydrodynamic control volumes containing core cells.

(type = integer/character*11, default = 0, units = none)

(4) ICORCV

CVH volume consistency switch.

(a) 0 or CONS
Consistency between fluid volumes in CVH and in COR, in the sense that the volume in COR may not exceed that in CVH, is required.

(b) 1 or NOCONS
Consistency between fluid volumes in CVH and in COR is not required.

(type = integer/character*6, default = 0, units = none)

Example

```
COR_MS  0 0 0 0 ! IEUMOD IHSDT IDTDZ ICORCV
```

COR_CMT – Candling Secondary Material Transport Parameters

Optional.

This record specifies the parameters needed to model the transport of secondary materials during candling. This record is not required, but if included, the following eight fields must be present. Transport mechanism 1 dictates that secondary material is relocated as a fraction of the molten mass specified by the transport parameter.

Transport mechanism 2 dictates that secondary material is relocated in fractional proportion, as specified by the transport parameter (with a value of 1.0 indicating direct proportionality), to its existing fraction within a component. For more detailed information on how these parameters are used in the candling model, see the COR Package Reference Manual, Section 3.1.

(1) MTUOZR
Transport mechanism flag for UO_2 in molten Zircaloy.
(type = integer, default = 1, units = none)

(2) MTZXZR
Transport mechanism flag for ZrO_2 in molten Zircaloy.
(type = integer, default = 1, units = none)

(3) MTSXSS
Transport mechanism flag for steel oxide in molten steel.
(type = integer, default = 2, units = none)

(4) MTCPSS
Transport mechanism flag for control poison material in molten steel.
(type = integer, default = 2, units = none)

(5) FUOZR
Transport parameter for UO_2 in molten Zircaloy.
(type = real, default = 0.2, units = none)

(6) FZXZR
Transport parameter for ZrO_2 in molten Zircaloy.
(type = real, default = 0.0, units = none)

(7) FSXSS
Transport parameter for steel oxide in molten steel.
(type = real, default = 1.0, units = none)

(8) FCPSS
Transport parameter for control poison material in molten steel.
(type = real, default = 0.0, units = none)

Example

```
!         MTUOZR MTZXZR MTSXSS MTCPSS FUOZR FZXZR FSXSS FCPSS
COR_CMT   1      1      2      2      0.20  0.00  1.00  0.00
```

COR_CCT – Component Critical Minimum Thicknesses

Optional.

This record specifies minimum thicknesses for cladding and other structures to maintain material intact (i.e., prevent material from becoming particulate debris). This record is not required, but if included, the following two fields must be present. Each value must be positive. The defaults are *ad hoc* values that should be varied in sensitivity studies to determine their impact on core degradation behavior. For more information on how these parameters are used in the debris formation model, see the COR Package Reference Manual, Section 3.2.1.

(1) DRZRMN

Critical minimum thickness of unoxidized metal (Zircaloy by default) in CL, CN, or CB. Set to 0.0 to allow uncollapsed, "bare" fuel pellets to survive to a temperature set by sensitivity coefficient 1132.

(type = real, default = 0.0001, units = m)

(2) DRSSMN

Critical minimum thickness of unoxidized metal (stainless steel by default) in SS, OS, SH, FM and RK.

(type = real, default = 0.0001, units = m)

Example

```
COR_CCT 1.0E-04 1.0E-04 ! critical min thickness
```

COR_BCP – Core Boundary Conduction Parameters

Optional.

This record specifies parameters for calculating conduction from the outer core ring to the boundary heat structures specified on record COR_ZP. This record is not required, but if included, the first four fields are required. For more information on how these parameters are used in the conduction model, see the COR Package Reference Manual, Section 2.2.7.

(1) ICBCD

Component number or name of component that conducts to boundary heat structures. From the list of component numbers (Section 1.1), only these components are permitted.

(a) CL
(b) CN
(c) CB
(d) SS
(e) NS
(f) FM
(g) RF

(type = Integer/character*2, default = none, units = none)

(2) MATBCD

Gap material for conduction to boundary heat structures.

(type = character*24, default = none, units = none)

(3) DXBCD

Gap thickness.

(type = real, default = none, units = m)

(4) CDFBCD

Boundary conduction thermal diffusion constant.

(type = real, default = none, units = m^2-K-$s^{1/2}$/J)

MELCOR 1.8.6 allows the user to use any gas defined in the NCG package or any material defined in the MP package as gap material. The data input for these materials will be used as default. If the user wishes to override or modify the default data, a tabular or control function may be specified on these optional fields which will then be used in calculation of the thermal conductivity.

(5) CFTFSW

An input switch specifying the interpretation of the function: TF for tabular function, CF for control function.

(type = character*2, default = none, units = none)

(6) CFTFNAM

Tabular or control function name (depends on the switch above) used for calculation of thermal conductivity.

(type = character*24, default = none, units = none)

Example 1

```
!     ICBCD MATBCD  DXBCD    CDFBCD   CF/TF FOR THC
COR_BCP  CN HELIUM  1.0E-04  2.0E-03    TF   TF222
```

Example 2

```
!     ICBCD MATBCD            DXBCD CDFBCD
COR_BCP  SS STAINLESS-STEEL   0.3   2.E-04
```

Example 3

```
!     ICBCD MATBCD  DXBCD      CDFBCD
COR_BCP SS H2O-VAP   1.0E-02   2.0E-04
```

COR_DX – Global Particulate Debris Exclusion Parameters

Optional.

This record allows both (1) global definition of volumes from which particulate debris will be excluded by the simple presence of various other components and (2) to define and/or redefine the global (or default) parameters of modeling for individual or several core cells falling within the range of axial level/radial ring numbers input on fields IA and IR respectively. The exclusion volumes are specified as fractions of the associated total volume, either channel or bypass. (Particulate debris will be excluded from the physical volume of the component even if this fraction is zero.) The model is described briefly in Sections 1.4 and 1.5.4 of this manual and in detail in Section 3.2.3 of the COR Package Reference Manual.

If this record is present, each data string may contain from four to nine fields. Default values will be retained for those fields that are not present and for any that contain -1.0. The default value for FBYXNS is different for a BWR than for a PWR. The choice is based on IRTYP input on record COR_RT.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) FCHXRD

Fraction of channel volume denied to particulate debris by presence of fuel rods, FU and/or CL.

(type = real, default = 1.0, units = none)

The following three fractions are used only in cells where there is a distinct bypass.

(5) FBYXRD

Fraction of bypass volume denied to particulate debris by presence of fuel rods, FU and/or CL.

(type = real, default = 0.0, units = none)

(6) FBYXCN

Fraction of bypass volume denied to particulate debris by presence of canister, CN.

(type = real, default = 0.3, units = none)

(7) FBYXCB

Fraction of bypass volume denied to particulate debris by presence of bladed canister, CB.

(type = real, default = 0.0, units = none)

The following two fractions are applied to the bypass volume in cells where there is a distinct bypass. Otherwise, they are applied to the total cell volume.

(8) FBYXNS

Fraction of applicable volume denied to particulate debris by presence of nonsupporting structure, NS.

(type = real, default = 0.7 for BWR, 0.0 for PWR, units = none)

(9) FBYXSS

Fraction of applicable volume denied to particulate debris by presence of supporting structure, SS.

(type = real, default = 0.0, units = none)

Examples

The following examples illustrate several forms of input for this record.

Suppose that NAXL=12 and NRAD=3.

This example defines the same constants for all core cells:

```
COR_DX 1 !IA   IR  FCHXRD FBYXRD FBYXCN FBYXCB FBYXNS FBYXSS
       1  ALL  ALL 1.0    0.0    0.3    0.0    0.7    0.0
```

Some equal variations of input are also possible:

```
COR_DX 2 !IA   IR  FCHXRD FBYXRD FBYXCN FBYXCB FBYXNS FBYXSS
       1  1-6  ALL 1.0    0.0    0.3    0.0    0.7    0.0
       2  7-12 ALL 1.0    0.0    0.3    0.0    0.7    0.0
```

or

```
COR_DX 3 !IA   IR  FCHXRD FBYXRD FBYXCN FBYXCB FBYXNS FBYXSS
       1  ALL  1   1.0    0.0    0.3    0.0    0.7    0.0
       2  ALL  2   1.0    0.0    0.3    0.0    0.7    0.0
       3  ALL  3   1.0    0.0    0.3    0.0    0.7    0.0
```

If the user redefines the global parameters for an individual or for several core cells, it should be kept in mind that definitions in later records take precedence over preceding records. Therefore, it is recommended that global definitions for all core cells occur first and that definitions for individual cells occur last. For example,

```
COR_DX 4 !IA   IR  FCHXRD FBYXRD FBYXCN FBYXCB FBYXNS FBYXSS
       1  ALL  ALL 1.0    0.0    0.3    0.0    0.7    0.0
       2  1-6  ALL 0.9    0.1    0.2    0.0    0.8    0.0
       3  ALL  1   1.0    0.0    0.3    0.0    0.7    0.0
       4  7    3   0.8    0.1    0.3    0.0    0.7    0.0
```

COR_BK – Global Core Blockage Enhancement Factor

Optional.

Changes in core geometry will be reflected as changes in flow resistance for those flow paths where an FL_BLK input record was included to invoke the flow blockage model.

This record allows both (1) global modification of the value used for connected porosity in evaluating the flow resistance from the Ergun equation and (2) to override the global (or default) definition of blockage enhancement for an individual or several core cells falling within the range of axial level/radial ring numbers input on fields IA and IR, respectively. If the user redefines the global parameters for an individual or several core cells it should be kept in mind that definitions in later records take precedence over preceding records. Therefore, it is recommended that global definitions for all core cells occur first and that definitions for individual cells occur last.

If this option is selected, the default value of porosity, which does not distinguish between internal and connected porosity, will be multiplied by the value of the specified control function (bounded between 0.0 and 1.0), but only for use in the blockage model.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ICFBKC

User name of control function to account for fraction of porosity in the channel that is connected. By default, or if the input is NO, the factor is assumed to be 1.0.

(type = character*16, default = none, units = none)

(5) ICFBKB

User name of control function to account for fraction of porosity in the bypass that is connected. If the input is NO, the factor is assumed to be 1.0.

(type = character*16, default = ICFBKC, units = none)

Example

This example defines the same constants for channel and bypass for all core cells:

```
COR_BK 1 !IA   IR  ICFBKC ICFBKB
       1  ALL  ALL  NO    CFblk
```

COR_NS – Global Support Rule for NS

Optional.

The primary use of the NS component in modeling a commercial reactor is to represent control structures (blades and rods). This record may both (1) define the global rule for support of NS in the core and (2) define and/or redefine the support rule of NS for an individual or several core cells falling within the range of axial level/radial ring numbers input on fields IA and IR, respectively. If the user redefines the global parameters for an individual or several core cells, it should be kept in mind that definitions in later records take precedence over preceding records. Therefore, it is recommended that global definitions for all core cells occur first and that definitions for individual cells occur last.

Note that BLADE and ROD options are available only for all core cells in a radial ring. The record is optional, and the default treatment depends on the reactor type (IRTYP on the COR_RT input record). If the record is input with less than the maximum permitted number of fields, default values will be used for the omitted field(s).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) INSSUP

Support rule for NS.

(a) 1 or BELOW
NS in a cell will collapse unless there is intact NS or unfailed SS in the cell immediately below it.

(b) 2 or ABOVE
NS in a cell will collapse unless there is intact NS in the cell immediately above it.

(c) 3 or FIXED
NS in a cell will not collapse until it fails locally.

(d) 4 or BLADE
Converted internally to BELOW for all cells.

(e) 5 or ROD
Converted internally to ABOVE for all cells except for the top axial level, where FIXED will be specified.

(type = character*6 default = BLADE for a BWR, and ROD for a PWR, units = none)

(5) METAL

Structural metal in NS.

(a) 1 or STEEL
Local failure is based on the thickness of steel in NS.

(b) 2 or ZIRC
Local failure is based on the thickness of Zircaloy in NS.

(type = character*6, default = STEEL, units = none)

(6) TNSMAX

Temperature above which NS will collapse, independent of remaining metal thickness. A zero or negative value implies collapse at the melting point of METAL.

(type = real, default = 0.0, units = K)

(7) DRNSMN

Critical minimum thickness of unoxidized metal in NS, below which it will collapse.

(type = real, default = 0.0001, units = m)

Example

```
COR_NS 1 !IA  IR  INSSUP  METAL  TNSMAX
       1  8   ALL BELOW   STEEL  1650.
COR_NS 1 !IA  IR  INSSUP  METAL  TNSMAX
       1  ALL 1   ROD     ZIRC   0.0
```

COR_SS – Global Loading and Failure Rule for SS

Optional

The primary use of the SS component in modeling a commercial reactor is to represent supporting structures, such as core plates and control rod guide tubes. This record may both (1) define the global rule for loading and failure of SS and (2) define and/or redefine modeling of SS for an individual or several core cells falling within the range of axial level/radial ring numbers input on fields IA and IR, respectively. The options are described briefly in Section 1.3 of this report, and in more detail in the COR Package Reference Manual.

In MELCOR 1.8.6, the lowest element of a column is treated as self-supporting; it will not collapse until it itself fails.

The record is optional. The field ISSMOD defines the structural model, while the fifth and subsequent fields provide parameters for the failure calculation, which may be either parametric or stress based. The default treatment is as a grid supported plate (PLATEG) with failure on over-temperature. If the record is input with less than the maximum permitted number of fields for the model specified in the ISSMOD field, default values will be used for the omitted field(s).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

Radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ISSMOD

Structural model option for SS.

(a) PLATE
SS in a cell will be treated as an edge-supported plate.

(b) PLATEG
SS in a cell will be treated as a grid-supported plate.

(c) PLATEB
SS in a cell will be treated as a BWR core plate.

(d) COLUMN
SS in a cell will be treated as a column usually representing a BWR control rod guide tube. The bottom of a column (SS is defined as a column in a cell with no column in the cell below) is treated as self-supporting unless there is an SS defined as a PLATE or PLATEG in the cell below. If either type of plate is present, it is assumed to support the bottom of the column, so that the column will collapse when the plate fails. Otherwise, the bottom of the column will not collapse until it itself fails.

(e) ENDCOL
SS in a cell will be treated as a self-supported element of a column and will not collapse until it itself fails. (The lowest element of a column that does not rest on a PLATE or PLATEG is internally converted to ENDCOL.)

(f) NAME
User-defined SS type with properties defined on a COR_UDSS input record.

(type = character*6, default = 'PLATEG', units = none)

For parametric models, the fifth field is a character variable specifying the model, and the interpretation of the sixth field depends on the model specified.

(5) ISSFAI

Failure model, either

(a) 1 or TSFAIL,
(b) 2 or LOG-CF.

(type = character*6, default = TSFAIL, units = none)

followed by either

(6) TSSFAI

Failure temperature for the TSFAIL model.

(type = real, default = 1273.15, units = K)

or

(6) ISSLCF

Name of a logical control function for the LOG-CF model. A .TRUE. value will define failure of the SS.

(type = integer, default = none, units = none)

For parametric failure models, an optional seventh field is available on which the user can specify whether the "structural" metal (the one checked for failure by loss of material) is steel , zircaloy or graphite (graphite is used in the HTGR reactor type).

(7) SSMETAL

Name of structural metal; must be STEEL, ZIRC or GRAPH

(type = character, default = STEEL, units = none)

For stress-based failure modeling, fields 5 through 7 define the parameters to be used in converting total loads borne by a structure. Their interpretation depends on the structural model option specified in field ISSMOD.

For the PLATE model, the stress at radius r in a plate of thickness h and radius a with a central hole of radius r_0, bearing a total load W_{total} is computed as

$$\sigma_e(r;r_0) = 6\,K_0 \frac{1}{\pi}\frac{1}{h^2}\left[1 - K_1\left(\frac{r}{a}\right)^2\right]\left[1 + \left(\frac{r_0}{r}\right)^2\right] W_{total} \qquad \text{(PLATE)}$$

The input parameters are

(5) THICK

Working thickness of plate, h, used only for stress calculations.

(type = real, default = none, units = m)

(6) AKM0

Lead coefficient, K_0.

(type = real, default = 0.206, units = none)

(7) AKM1

Coefficient K_1 in stress concentration term for central hole.

(type = real, default = 0.576, units = none)

For the PLATEG model, the maximum stress in a plate of thickness h supported by beams of spacing h with a total load, W_{total} on an area A_{ring} is computed as

$$\sigma_{e,ring} = 6\, K_G \frac{x^2}{A_{ring}} \frac{1}{h^2} W_{ring} \qquad \text{(PLATEG)}$$

The input parameters are

(5) THICK

Working thickness of plate, h, used only for stress calculations.

(type = real, default = none, units = m)

(6) SPACE

Spacing of supporting beams, x.

(type = real, default = none, units = m)

(7) AKMG

Coefficient K_G.

(type = real, default = 0.0513, units = none)

For the PLATEB model, the maximum stress in a plate of thickness h supported by beams of spacing h with a total load, W_{total} on an area A_{ring} is computed as

$$\sigma_{e,ring} = 6\, K_B \frac{x^2}{A_{ring}} \frac{1}{h^2} W_{ring} \qquad \text{(PLATEB)}$$

The input parameters are

(5) THICK

Working thickness of plate, h, used only for stress calculations.

(type = real, default = none, units = m)

(6) SPACE

Spacing of supporting beams, x.

(type = real, default = none, units = m)

(7) AKMB

Coefficient K_B.

(type = real, default = 0.0833, units = none)

For the COLUMN model, the maximum stress in a group of N identical cylindrical columns with initial outer radius r_o and inner radius r_i with a total load, W_{total} is computed as

$$\sigma_{e,ring} = \frac{1}{N\pi\left(r_o^2 - r_i^2\right)} \frac{M_{column,0}}{M_{column}} W_{ring} \qquad \text{(COLUMN)}$$

The factor of the ratio of the initial column mass to the current mass is included to account for any reduction of load bearing area by oxidation or melting. The input parameters are

(5) XNUM

Number of identical columns in ring, N

(type = real, default = none, units = none)

(6) RADO

Outer radius of column, r_o.

(type = real, default = none, units = m)

(7) RADI

Inner radius of column, r_i.

(type = real, default = 0.0, units = m)

For ISSMOD = PLATE or PLATEG only, three additional parameters may be used to add the pressure differential loading or a logical control function to fail the structure. The input parameters are

(8) CFP

Name of CF to apply optional pressure loading and/or head loading, or the word NO. This is optional; NO is accepted in case CFR is used and CFP is not.

(type = character*16, default = none, units = none)

(9) AREAR

Area over which to apply pressure/head loading calculated by control function CFP. Note that if CFP = NO, this is ignored but must be present.

(type = real, default = none, units = m^2)

(10) CFR

Name of logical CF to be read with the stress-based failure calculation. CFR should return .TRUE. when a failure condition is met, .FALSE. otherwise. This can be left off if not used.

(type = character*16, default = none, units = none)

Example

```
COR_SS   2 ! IA IR ISSMOD ISSFAI   TSSFAI
         1   5 ALL PLATE  TSFAIL   600.
         2   6   3 PLATE  TSFAIL   500.
```

For the PLATEB model, the maximum stress in a plate of thickness h supported by beams of spacing h with a total load, Wtotal on an area Aring is computed as

COR_UDSS – Properties of User-Defined SS Type

Optional.

The user may define one to three types of SS in addition to the built-in types. Each is assigned a name, and its support capabilities are defined by input on a string of the COR_UDSS tabular record. Until failure, all user-defined types are currently assumed to be self-supporting, and the ability to support other intact components and/or particulate debris is defined by user input. The fate of the SS itself after failure is also defined by user input.

The name of a user-defined SS type can be used on the COR_SS input record in the same way as built-in types, such as PLATE and COLUMN. However, only parametric failure options (TSSFAI or ISSLCF) are allowed.

(1) NUMUDS

Number of user-defined SS types (from one to three).

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMUDS:

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) CSSUDF

Name of user-defined SS type.

(type = character*6, default = none, units = none)

Support capabilities are defined by keywords in one or more of fields 3 through 5. If a keyword is absent, the corresponding capability is assumed to be absent. There is no dependence on the order of keywords, but at least one of them must be present.

(3-5) CSSOPT

Support capability of user-defined SS type.

(a) 1 or INTACT
SS of type CSSUDF supports intact components in the cell above until failure.

(b) 2 or DEBRIS
SS of type CSSUDF supports all particulate debris (PD and PB) in the cell above until failure.

(c) 3 or SELF
SS of type CSSUDF continues to be self-supporting after failure.

(type = character*6, default = none, units = none)

Example

```
COR_UDSS   1 ! N CSSUDF CSSOPT
               1 'MYSUP'   'INTACT' 'SELF' 'DEBRIS'
```

COR_PC – Global Pool Heat Transfer from Bottom/Top of SS Plates

Optional.

This record may both (1) give the global definition of modeling of heat transfer from SS plate surfaces and (2) define and/or redefine modeling for the core cells falling within the range of axial level/radial ring numbers input on fields IA and IR, respectively.

Unlike other core components, plates have horizontal bottom and top surfaces. When the SS component is used to model a plate, heat transfer from these surfaces to water pools below and/or above may be calculated. The model may be controlled independently for the two surfaces and is off by default. When the model is on, the total area of either surface is taken as the total cross-sectional area of the core cell. The fraction covered below is ramped on as the surface of the pool in the core cell below rises to the bottom surface of the plate. Similarly, the fraction covered above is ramped off as the surface of the pool in the core cell above falls to the top surface of the plate.

By default, the heat transfer coefficient for the bottom surface is evaluated from the built in correlation for downward-facing boiling and that for the top surface from the built-in pool boiling correlation. These may be overridden by constant values or by values calculated as control functions. In any case, the temperature difference is based on the average temperature of the plate in the core cell and that of the pool.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DZBOT

Clearance between bottom of plate and pool surface for no contact.

If negative, no heat transfer is considered between the bottom surface and the pool.

(type = real, default = -1.0, units = m)

(5) DZTOP

Pool depth over top surface of plate for complete coverage.

If negative, no heat transfer is considered between the top surface and the pool.

(type = real, default = -1.0, units = m)

(6) IHTCBOT

Key for heat transfer coefficient between bottom surface and pool.

(a) CONST
Use constant value.

(b) CF
Use control function.

(type = character*5, default = CONST, units = none)

optionally followed by either

(7) HPBOT

Constant heat transfer coefficient between bottom surface and pool to override the default boiling correlation.

(type = real, default = 0.0, units = W/m^2K)

or

(7) IHPBOT

Name of a real control function whose value will be used as the heat transfer coefficient between bottom surface and pool.

(type = character*16, default = none, units = none)

(8) IHTCTOP

Key for heat transfer coefficient between top surface and pool.

(a) CONST
Use constant value.

(b) CF
Use control function.

(type = character*5, default = CONST, units = none)

optionally followed by either

(9) HPTOP

Heat transfer coefficient between top surface and pool to override the default boiling correlation.

(type = real, default = 0.0, units = W/m^2K)

or

(9) ICFEMI

Name of a real control function whose value will be used as the heat transfer coefficient between top surface and pool.

(type = character*16, default = none, units = none)

Example

```
COR_PC   1 !IA IR  DZBOT  DZTOP  CONST/CF HPBOT  CONST/CF HTBOT
          1  2  3   -1.   -1.   CONST    0.    CONST    0.
```

COR_PR – Global Downward Radiation from SS to Pool or Lower Head

Optional.

This record may both (1) give the global definition of modeling of downward radiation from SS and (2) override modeling for the core cells falling within the range of axial level/radial ring numbers input on fields IA and IR, respectively.

By default, the general radiation model is used for radiation from the lowest surface in the core to a pool or the lower head below. The view-factor-times-area product is taken as FLPUP MIN(A_{surf}, ASCELA) or FCELA MIN(A_{surf}, A_{LH}), respectively. Here FLPUP and FCELA are values input on the COR_RF record (or associated defaults), ASCELA is input for each ring on COR_RP record, and A_{LH} is derived from input on COR_LHD

record. In some cases, this may not adequately represent actual radiative heat transfer from the core support plate. This record allows generalization of the calculation. It affects only the specific case of radiation from SS as the lowest surface in the core to a pool or the lower head below. The input parameters are

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ISSVFA

Name of a real control function whose value (subject to appropriate limits) will be used as the view-factor-times-area product for this radiation path.

(type = character*16, default = none, units = none)

(5) KEYEMIS

Key for emissivity value to be used for the SS surface.

(a) CONST

Use constant value.

(b) CF

Use control function whose value will be used as the emissivity.

(type = character*5, default = CONST, units = none)

optionally followed by either

(6) EMISSS

Constant emissivity value to be used for the SS surface.

(type = real, default = 0.0, units = none)

or

(6) ICFEMI

Name of a real control function whose value (subject to appropriate limits) will be used as the emissivity of the SS surface.

(type = character*16, default = none, units = none)

Example

```
COR_PR   1 !IA  IR       ISSVFA   CONST/CF EMISS
          1 2    3       CF100    CF        CF101
```

COR_CR – Control Rod Silver Release

Optional.

This record specifies parameters for the control rod silver release model. It can be used if the control rod poison used is Ag-In-Cd. This model allows the release of degraded Ag-In-Cd fuel rod material to the RadioNuclide (RN) package. The default behavior, without the model, would be for control rods to fail and control rod poison to relocate as a conglomerate; however, the relocated alloy would not be released to the RN package as aerosol or vapor.

In conjunction with the model, three classes must be defined in DCH with specific names: AG-CR, IN-CR, and CD-CR. These classes will hold the nonradioactive masses of the alloy components released from the control rods. Sensitivity coefficients C7110, C7111, and C7120 in RN must also be defined for these classes in order to set the vapor pressure, diffusion coefficients, and molecular weights, respectively.

The input parameters are

(1) IAICON

Turns on the silver release model

(a) 0 or NACT
Model is not active. No additional fields are required.

(b) 1 or ACTC
Model is active, vaporization is allowed from candling material only

(c) 2 or ACTDC
Model is active, vaporization is allowed from both candling material and conglomerate

(type = integer / character*5, default = 0, units = none)

The following fields are required if and only if IAICON = ACTC or 1, ACTDC or 2. If they are not set, then the default values will be used:

(2) ARATIO

The area ratio of break area to control rod internal cross-sectional area. This cannot be greater than 1.

(type = real, default = 0.1, units = none)

(3) AKFRCT

The flow loss coefficient used in the release velocity calculation. If input, this will override the value calculated according to the formula given in the reference manual.

(type = real, default = 0.32*(1.0 - ARATIO), units = none)

Example

```
COR_CR NACT
```

COR_B4C – Control Rod B_4C Oxidation Model

Optional.

This record specifies parameters for the control rod B_4C oxidation model. It can be used if the type of control rod poison used is B_4C. The default behavior without the model is that of the old model: the simple B_4C oxidation chemistry and the old release/oxidation behavior are in effect. This means that total B_4C oxidation is limited to 0.02 [default on C1031(2)], oxidation will start when the control rod clad fails, and control rod material that is released (failed) goes into the COR particulate field. The parameters on C1031 hold, except for C1031(1), the former simple/advanced oxidation switch, which is inactive. The CORB4C record activates both the advanced boron chemistry and the new control rod release/oxidation model.

All fields on this record are optional. The input parameters are

(1) BCSA

Selects either the simple or advanced boron chemistry.

(a) 0 or SIMP
Simple boron chemistry model.

(b) 1 or ADV
Advanced boron chemistry model. Note that the usual input to DCH (to set up the boron RN classes) is necessary.

(type = integer / character*4, default = SIMP, units = none)

(2) BCMOD

Selects either the original failure model or the new model.

(a) 0 or BLADE
Old model, failure/oxidation at clad failure temperature, released B_4C goes to the particulate field.

(b) 1 or ROD
New model, oxidation begins at TBCOX. When clad failure occurs, the B_4C will candle. Note that the B_4C material properties must be altered using MP input to allow this (see below).

(type = integer / character*5, default = 'BLADE', units = none)

(3) TBCLIQ

The BC liquefaction temperature. This is where the B_4C/SS liquefies – melting point of B_4C should be set to this in MP if the ROD model is used.

(type = real, default = 1700, units = K)

(4) TBCOX

The B_4C oxidation temperature. If the ROD model is used, oxidation starts at this temperature whether the control rod clad has failed or not. Typically set to the B_4C/SS eutectic formation temperature.

(type = real, default = 1500, units = K)

Example

```
COR_B4C SIMP  BLADE  1650.0  1450.0
```

If the ROD model is used, the B_4C melting point and enthalpy table should be modified via MP input. Although the default B_4C control material properties can be altered via MP and tabular functions (TF), a simpler way to do this is to change the control material to B4C-INT (B_4C Interactive) on the COR_RT record (MCRP).

The default melting point for this material is 1700 K, and the heat of fusion is the same as for the normal B_4C. Of course, if a different liquefaction point is entered for TBCLIQ above, the B4C-INT melt temperature should be altered to match via MP input. Note that because B4C-INT is an Interactive material, the enthalpy table cannot be altered via TF.

Use of the ROD model and the B4C-INT material (or modified BORON CARBIDE) allows the B_4C to be molten when the SS clad fails, thereby allowing the B_4C to candle down the control rod.

COR_ROD – Rod Collapse Model Record

Optional.

This record activates an additional framework for a fuel collapse model with sensitivity coefficients for residual lifetime failure criterion, i.e., time at temperature. This record is not required, but if it is included, at least one field must be present.

(1) IRODDAMAGE

Tabular function number for fuel failure time as a function of cladding temperature.

(type = integer, default = none, units = none)

(2) RCLADTHICKNESS

Minimum unoxidized clad thickness under which the rod collapse model supplants the default temperature based failure criterion.

(type = real, default = 1.0e-4, units = m)

Example

```
COR_ROD 900 1.0e-3
TF_ID     FUEL-FAIL-TIME       1.0         0.0
TF_TAB  4
*     (K)     (SECONDS)
1     1000.0  6.00E+31    * INFINITE AT 1000K
2     1001.0  3.60E+03    * 60 MIN AT 1001 K
3     2100.0  1.80E+03    * 30 MIN AT 2100 K
4     2500.0  6.00E+01    * 1 MIN IF > 2500 K
```

COR_OX – PSI Air Oxidation Model

Optional.

A new model for air oxidation and breakaway for Zircaloy-4 were developed by the Paul Scherrer Institute (PSI). The key features of the model are applicability to oxidation by both steam and air, and capturing the accelerated oxidation kinetics that occurs in air and also in steam under certain conditions. The details of this model are described more fully in the COR reference manual. This model is more theoretically based than the previous breakaway model (see COR_OXB record) which provides a parametric means for determining breakaway.

As part of this new model, alternate models for oxidation of Zircaloy-4 in steam were also included in this model. The input parameters are

(1) MODEL

Key for activation of PSI oxidation model

(a) 0 or MELCOR
Previous MELCOR oxidation model is used

(b) 1 or PSI
PSI oxidation model is used

(type = integer or character, default = 0, units = none)

The following data must be provided if, and only if MODEL = 1:

(1) STEAM

Steam oxidation model

0 - Catchart-Pawel/Urbanic-Heidrick;
1 - Leistikov-Schanz/Prater-Courtright;
2- Leistikov;
3- Urbanic-Heidrick;
4 - Sokolov;
5 - Grosse;
-1 - Kinetic parameters given by sensitivity cards (SC1001)
(type = integer, default = none, units = none)

(2) AIR

Air Oxidation Model

0 – Hofmann-Birchley;
1 – Hayes-Roberson/Leistikov-Berg (NUREG1);
2- Powers (NUREG2) (Birchley);
3- Benjamin (MELCOR 1.8.6) (Birchley);
4 – Mozart (Birchley);
-1 – Kinetic parameters given by sensitivity cards (SC1001);
(type = integer, default = none, units = none)

OXYGEN

Oxygen oxidation model

0 – Uetsuka-Hofman;
-1 – Kinetic parameters given by sensitivity cards (SC1001);
(type = integer, default = none, units = none)

NOBRK

Breakaway Switch

0 – enable in steam and air (recommended);
1 – enable in air; disable in steam;
2- disable in both steam and air;
3- enable in steam, disable in air;
4 – enable in steam and air, apply the steam breakaway parameters also in air (i.e. air is like steam)
5 - enable in steam and air, apply the air breakaway
6 – enable in air, disable in steam if not already in breakaway
7 – enable in steam, disable in air if not already in breakaway
8 – enable in steam, disable in air if pre--oxidised in steam and not already in breakaway from steam

(type = integer, default = none, units = none)

Example

```
COR_OX 1  0  0  0  0
```

COR_OXB – Zircaloy Air Oxidation Breakaway model

Optional.

This model has been implemented in MELCOR by adding an optional breakaway lifetime calculation to the oxidation model. This is a parametic model for predicting the onset of breakaway. The PSI air oxidation model (COR_OX) provides a more mechanistic way of modeling air oxidation of Zircaloy. This model is activated by the input record COR_OXB, along with sensitivity coefficients 1016 and 1017. The model calculates the lifetime for Zircaloy components in each cell, as specified by the control record. The new post-breakaway oxidation coefficients are supplied in 1016. This has the same structure as the original Zircaloy coefficients in 1001, except that there are no transition temperatures and only one set of coefficients, for post-breakaway. To use the breakaway model, the 1001 coefficients should have both the low and high temperature coefficients for air set to pre-breakaway values, or, alternatively, set the low-temperature coefficients and set the transition temperatures to very high values.

The second sensitivity coefficient 1017 contains the fit coefficients for the lifetime rule.

The post-breakaway rate coefficients will be used when the lifetime of a Zircaloy component in a cell (as set on the COR_OXB record) becomes 1. The lifetime is calculated as (see sensitivity coefficient 1017 below)

$$LF = \int_0^t dt' \frac{t'}{\tau(T)}$$

The input parameters are

(1) IOXB

Oxidation model option.

(a) 0

Breakaway model is off.

(b) 1

Breakaway model is on for clad and canister.

(c) 2

Breakaway model is on for all Zircaloy components (not recommended).

(type = integer, default = none, units = none)

Example

```
COR_OXB 1
```

COR_TKE – Core Effective Thermal Conductance

Optional for PBR and PMR.

This is a new record recognized for the PBR and PMR COR types that can be used to specify the effective bed conductance (PBR) or radial core conductance (PMR). This model has been implemented in MELCOR by adding a Zehner-Schlunder-Bauer model for pebble bed thermal conductance and a Tanaka-Chisaka porosity model for the block radial conductivity model in PMRs.

The input parameters are

(1) TKETF

Name of tabular function to be used for effective radial bed conductivity. If 'NO' is entered, no TF is used. If, present, the tabular function will be used instead of the internal conductivity models.

(type = character, default = '', units = none)

(2) PORCHAN

Effective block porosity to use in the Tanaka-Chisaka model for effective radial conductivity for PMR.

(type = real, default = 0.0, units = none)

(3) DBLK

Effective size of the graphite block for the gap model for effective radial conductivity of PMR.

(type = real, default = 1.0, units = m)

(4) BLKGAP

Gap width between graphite blocks for the gap model for effective radial conductivity of PMR.

(type = real, default = 0.0, units = m)

Example

```
COR_TKE 'TK_BED'
```

COR_PKM01 – Reactor Point Kinetics Model (1)

Optional

This record specifies parameters for the reactor point kinetics model. The existence of this record activates the point kinetics model.

The input parameters are

(1) Tinit

Initialization time for point kinetics model. This value determines the time at which the point kinetics model becomes active.

(type = real, default = none, units = seconds)

(2) Qinit

Initialization power for point kinetics model. This value determines the initial power level for the model and the constant power used until the point kinetics model becomes active.

(type = real, default = none, units = W)

(3) Fuel

Material used for fuel within the point kinetics model. This material (i.e., Zr, Inc) is used in the temperature feedback reactivity equations.

(type = character, default = none, units = none)

(4) Moderator

Material used for moderator within the point kinetics model. This material (i.e., Zr, Inc) is used in the temperature feedback reactivity equations.

(type = character, default = none, units = none)

Example

```
COR_PKM01   0.0   1.68E6   Zr   Inc
```

COR_PKM02 – Reactor Point Kinetics Model (2)

Optional

This record specifies parameters for the reactor point kinetics model.

The input parameters are

(1) ExtReac

Control function name for external reactivity. Control function must provide reactivity in units of dollars.

(type = character*16, default = none, units = none)

(2) NeuSrce

Control function name for neutron source. Control function must provide neutron source in units of neutrons/second.

(type = character*16, default = none, units = none)

Example

```
COR_PKM02   'CF-REACTIVITY' 'CF-N-SOURCE'
```

COR_PKM03 – Reactor Point Kinetics Model (3)

Optional

This record specifies what CF to use for the feedback functions for the reactor point kinetics model.

The input parameters are

(1) NN

Number of table lines.

(type = integer, default = none, units = none)

The following data are input as a table with length NN:

(1) N

Line index.

(type = integer, default = none, units = none)

(2) PKname

The name of a CF to use for this region and component, or a default equation. The defaults are the present feedback equations plus the Xe feedback:

(a) RDOPF = Doppler feedback equation
(b) RFUF = Fuel density feedback equation
(c) RMODF = Moderator density feedback equation
(d) RXEF = Xenon feedback equation

(type = character*16, default = none, units = none)

PKname is given here because we have to know which CF or equation to hook to the region average temperature, given by record COR_TAVG. The user-input CFs and the Xe feedback function are not fully implemented yet, so operation is restricted to the first three built-in functions.

(3) ITAVG

number of line in COR_TAVG table to use for average temperature.

Example

```
COR_PKM03  3
           1 RDOPF 1
           2 RFUF  1
           3 RMODF 2
```

COR_TAVG - Region Average Component Temperature

Optional.

This record specifies the core region to use for averaging component temperatures. These temperatures are for use in the point kinetics model, but since they are CF variables, they could be used in other parts of MELCOR. (NOTE - not available yet as CF variables).

(1) NN

Total number of lines in table.

(type = integer, default = none, units = none)

The following data are input as a table with length NN:

(1) N

Line index.

(type = integer, default = none, units = none)

(2) IA

Axial core cell region (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

Radial core cell region (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) COMP

The name of a COR component to use for temperature and mass-weighting. The components are:

(a) FU

Fuel component

(b) MOD

Moderator, which is actually the CL component in a HTGR

(c) RF

Reflector, which is internally the CB component

(5) MAT

Name of material in COR component to be used for temperature and mass-weighting. The materials are the usual: UO2, ZR, GRAPH, etc.

Note that the above names now refer to a component and material, not a material as before. For instance, previously FU referred to the UO2; now, it refers to the fuel particles + matrix, which makes up the FU component in MELCOR.

Example

```
COR_TAVG  2
           1 ALL  ALL FU  UO2
           2 6-15 2-4 MOD GRAPH
```

COR_DIFF - Steady State Diffusion

Optional.

This record specifies data to start the steady state operation fission product diffusion calculation for the HTGR FP model.

(1) SSFLAG

Activates the SS diffusion calculation.

(type = integer, default = 1, units = none)

(2) TIMEON

MELCOR problem time when the calculation is started.

(type = real, default = 0.0, units = s)

(3) MDIFFILE

The name of the file containing the model setup data.

(type = character, default = '', units = none)

(4) INITFILE

The name of an output file containing distribution and release data to initialize the FP transport problem.

(type = character, default = 'Init.out', units = none)

During the SS calculation, the MELCOR timestep is not advanced. After the SS calculation, control returns to MELCOR.

Example

```
COR_DIFF 1 0.0 'mdif.in' 'Init.out'
```

COR_XPRT - Transport HTGR FP Model

Optional.

This record starts the transport phase of the HTGR steady state operation.

(1) TIMEON

The MELCOR problem time to start the calculation.

(type = real, default = 0.0, units = s)

(2) INITFILE

The filename with the diffusion initial conditions from a previous COR_DIFF run.

(type = character, default = 'Init.out', units = none)

(3) TRANSFILE

The filename to write out the initial conditions generated to start a follow-on transient calculation.

(type = character, default = 'Trans.out', units = none)

As now implemented, the transport calculation is done from TIMEON to TEND, the problem end time, and TRANSFILE is written at TEND. TRANSFILE contains the RN class FP distributions in the CVs, deposited on the HSs, and deposited in the filters. Note that COR_XPRT is used with a mapping of tracked isotopes from the SS diffusion calculation to RN classes provided by RN1_DIFMAP.

Example

```
COR_XPRT 0.0 'Init.out' 'Trans.out'
```

COR_DIFT - Start HTGR FP Release Model Transient

Optional.

This record initializes a transient mode HTGR FP release model, using input data from previous SS diffusion and transport calculations, along with a failure fraction input.

(1) TIMEON

The MELCOR problem time when the calculation is started (unneeded for a new problem).

(type = real, default = 0.0, units = s)

(2) GSCALE

Global scaling factor to be applied to the deposition data on TRANSFILE to scale the length of time of the transport problem calculation to the actual desired operation time.

GSCALE can be overridden for specific RN classes using the RN1_DSCAL record.

(type = real, default = none, units = s)

(3) INITFILE

The filename with the diffusion initial conditions from a previous COR_DIFF run.

(type = character, default = 'Init.out', units = none)

(4) TRANSFILE

The filename with the initial conditions from a previous COR_XPRT run.

(type = character, default = 'Trans.out', units = none)

(5) MDIFFILE

The name of the file containing the FP model setup data.

(type = character, default = 'mdif.in', units = none)

(6) FAILFFILE

The filename for the failure fraction surface to be used during the transient calculation. This can also be NONE, CF, or TF. If CF/TF, followed by the CFNAME/TFNAME. The current release version only uses NONE, which specifies an internal failure fraction fit to AVR failure data.

(type = character, default = 'NONE', units = none)

2.2 MELGEN Axial Level Input

COR_ZP – Axial Level Parameters

Required.

This record specifies various data for all axial levels:

(1) NAXL

Total number of axial levels in the core and lower plenum.

(type = integer, default = none, units = none)

The following data are input as a table with length NAXL:

(1) IA

Axial level number.

(type = integer, default = none, units = none)

The following fields specify various geometric parameters for each axial level. These three floating point fields are required for each level:

(2) Z

Elevation of lower boundary of axial level.

(type = real, default = none, units = m)

(3) DZ

Axial length of the level from lower boundary to upper boundary. For all but the uppermost elevation, DZ can be calculated by MELGEN from the difference in Z specified for bounding elevations. The user can specify a hyphen, '-', to indicate that MELGEN should calculate this difference for any elevation but the top.

(type = real, default = none, units = m)

(4) PORDP

Porosity of particulate debris for all cells in the given axial level IA. The value must be nonnegative and less than 1.0.

(type = real, default = none, units = none)

For axial level IA above the elevation of the BWR baffle plate or the PWR bottom plate input as HLST on record COR_AVP, the fifth field specifies the heat structure name representing the radial thermal boundary and is required for each level. In most cases, heat structures above this level will represent a portion of a BWR core shroud or a PWR core support barrel.

In MELCOR 1.8.6, the lower-head model is used to treat that portion of the vessel that is below HLST. Input of IHSA is still required, and IHSA input must be a character string NO.

Melting of these structures can be modeled through appropriate input to the HS package, using HS_DG input records (see the HS Package Users' Guide). When this is done, the molten steel from the structure will be added to the outermost radial ring of the axial level containing the structure.

(5) IHSA

Boundary heat structure name for this axial level, or NO for levels below HLST. A unique heat structure must be specified for each axial level above HLST.

(type = character*16, default = none, units = none)

The sixth field on this record specifies the relative amount of fission power generated per unit fuel mass in the given axial level IA, as well as the relative amount of decay power per unit mass if the RN package is inactive. The sum of the values for all levels must be greater than 0.0 if ICFFIS on input record COR_TP is not 0 or if the RN package is inactive. The values are normalized for all cells containing a fuel component

for ICFFIS = ±1, or (for fission power only) for all liquid-covered cells containing fuel for ICFFIS = ±100. This field is optional.

(6) FZPOW

Relative power density in this level.

(type = real, default = 1.D0, units = none)

A seventh field on this record can be used to specify the failure temperature for PWR core formers (FM). This field is optional; if present, the sixth field must be present, at least as a place-holder.

(7) FMRTFL

Temperature at which PWR core formers (FM) will fail.

(type = real, default = 1073.15, units = K)

Example

```
COR_ZP 5 !IAXL   Z     DZ       PORDP     IHSA      FZPOW   FMRTFL
          1      5.9   0.1      0.8999     NO       0.1
          2      6.0    -       0.8999     HS11003  0.1     1300.  !MELGEN calculates DZ
          3      7.0   1.0      0.8998     HS11004
          4      8.    1.0      0.8998     HS11005
          5      9.    1.0      0.8998     HS11006
```

COR_SHS – PWR Core Baffle (Shroud) Support Options

Optional

This record specifies the mechanism assumed for support of the core baffle (shroud) in a PWR in one or more axial levels.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as a–b [without any blanks] or the character string ALL).

(type = integer, default = none, units = none)

The third field specifies the support option for that level or levels:

(3) Z

Support option; must be FORMERS or FIXED.

(type = character, default = FORMERS, units = none)

Example

```
COR_SHS   2 ! n IA  Option
              1 3-7 FORMERS
              2 9   FIXED
```

2.3 MELGEN Radial Ring Input

COR_RP – Radial Ring Parameters

Required.

This record specifies various data for all radial rings.

(1) NRAD

Number of radial rings in the core and lower plenum.

(type = integer, default = none, units = none)

The following data are input as a table with length NRAD:

(1) IR

Radial level number.

(type = integer, default = none, units = none)

The second field specifies the outer radius of each ring. The values are used for overall nodalization of core radial rings. It is required that the values of RCOR and RVESS, input on record COR_VP, be included in the list of ring radii.

The input radii also allow the definition of default values for other input variables, such as ASCELA, for which input was previously required. Note that these variables may still be input independently and that there is no requirement that input values be consistent with cylindrical geometry based on the ring radii.

(2) RINGR

Outer radius of ring IR in COR nodalization.

(type = real, default = none, units = m)

The third field specifies the total axial cross-sectional area of each radial ring, including all core components, channel flow area, and bypass flow area. This value is used in the radiation model for calculating axial inter-cell radiative energy exchange. It is not used

in calculating core cell volumes. These are based on component volumes and fluid flow areas entered on the COR_BFA record. This floating point field can be set for each ring. For further details, see the COR Package Reference Manual, sections 1.1 and 2.1.

(3) ASCELA

Total cross-sectional area of this radial ring IR.

Optional field.

(type = real, default = π ($R_o^2 - R_i^2$), where R_o and R_i are the outer and inner radii of ring IR, units = m^2)

If the input value differs significantly from the default value based on ring radii, the input is accepted but a WARNING is issued by MELGEN.

The next field is either the fourth or third, depending on whether ASCELA is present. This field specifies the heat structure name representing the upper axial boundary for radial ring IR. This field is required for each level. Melting of these structures can be modeled through appropriate input to the HS package, using HS_DG input records. When this is done, the molten steel from the structure will be added to the outermost radial ring of the uppermost axial level of the core. This field is followed by a placeholder corresponding to the field ICFLHF which was moved to the COR_PEN record.

(3 or 4) IHSR

Upper boundary heat structure name for this radial ring.

(type = character*16, default = none, units = none)

(4 or 5) UNUSED

This field is currently unused. This field was previously used to specify a control function that could specify penetration failure. This field was moved to the COR_PEN record. Enter '-' as a placeholder.

The following two fields allow the use of control functions to specify the direction of flow to be assumed in the oxidation calculation, and the order in which core cells within a ring will be given access to available oxidizers. If a name is entered, the direction will be inferred from the value of the specified control function, with a positive value taken to imply upflow and a negative value to imply downflow. These fields are optional but if the user wants to set the last optional field (the eighth—FRPOW), a character string NO is required as a placeholder for both fields, but will not be used.

(5 or 6) ICFCHN

The name of a REAL control function from which the direction of flow in the channel will be inferred.

Input NO if control function is not used.

Optional field.

(type = character*16, default = NO, units = none)

(6 or 7) ICFBYP

The name of a REAL control function from which the direction of flow in the bypass will be inferred.

Input NO if control function is not used.

Optional field.

(type = character*16, default = NO, units = none)

The last field specifies the relative amount of fission power generated per unit fuel mass in radial ring IR, as well as the relative amount of decay power per unit mass if the RN package is inactive. The sum of the values for all rings must be greater than 0.0 if ICFFIS on input record COR_TP is not 0 or if the RN package is inactive. The values are normalized for all cells containing a fuel component for ICFFIS = ±1, or (for fission power only) for all liquid-covered cells containing fuel for ICFFIS = ±100.

(7 or 8) FRPOW

Relative power density in this ring.

Optional field

(type = real, default = 1.0, units = none)

Example

```
COR_RP  3 ! NRAD
! IR  RINGR         ASCELA  IHSR     Unused  ICFCHN  ICFBYP  FRPOW
  1   1.15                  HS15001  -       NO      NO
  2   1.80                  HS15002  -       NO      NO
  3   2.20                  HS15003  -       NO      NO
```

2.4 MELGEN Specific Cell Input

COR_RBV – Cell Reference and Fluid Boundary Volumes

Required.

This record specifies a reference cell for defining missing input and the fluid control volume name(s) adjacent to each cell.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

If a core cell lies in the inaccessible portion of the core nodalization, so that it contains no core components and no fluid volume and will never contain any, MELCOR 1.8.6 allows the complete specification of this cell on the COR_RBV record by use of the single keyword NULL in the field that would otherwise contain the number of the reference cell.

(4) IREFN

NULL to specify that a core cell lies in the inaccessible portion of the core nodalization.

(type = character*4, default = none, units = none)

In this case, no other input records for the NULL cell are required and, in fact, none will be read.

If the reference cell (IREF, JREF) is input, any input values that are missing from the input for the given cell (IA, IR) are set to the values for cell (IREF, JREF) and no input error is generated. These fields (4) and (5) may be set to 0 if no reference cell is needed, but then all other input for this cell must be included.

(4) IREF

The axial level number for reference cell for setting values not input for this cell.

(type = integer, default = 0, units = none)

(5) JREF

Radial level number for reference cell for setting values not input for this cell.

(type = integer, default = 0, units = none)

(6) ICVHC

Channel control volume adjacent to this cell. This field may be missing or set to REF if a valid reference cell is input on fields (IREF, JREF).

(type = character*16, default = REF, units = none)

(7) ICVHB

Bypass control volume adjacent to this cell. For a BWR, this represents the interstitial volume between canister boxes; for a PWR, it represents the volume in the barrel/baffle region. The field may be missing or set to REF if a valid cell is input for IREFN.

(type = character*16, default = REF, units = none)

Examples

```
COR_RBV 14 !IA  IR  IREF  JREF  ICVHC  ICVHB
         1  1   1   0     0     CV120  CV120
         2  1   2   0     0     CV120  CV120
         3  2   1   0     0     CV120  CV120
         4  2   2   0     0     CV120  CV120
         5  3   1   0     0     CV120  CV120
         6  3   2   0     0     CV120  CV120
         7  4   1   0     0     CV110  CV111
         8  4   2   0     0     CV110  CV111
         9  5   1   0     0     CV110  CV111
        10  5   2   0     0     CV110  CV111
        11  6   1   0     0     CV110  CV111
        12  6   2   0     0     CV110  CV111
        13  7   1   0     0     CV110  CV111
        14  7   2   0     0     CV110  CV111
```

or

```
COR_RBV  2 !IA  IR  IREF  JREF  ICVHC  ICVHB
         1  1-3 ALL 0     0     CV120  CV120
         2  4-7 ALL 0     0     CV110  CV111
```

or

```
COR_RBV  6 !IA  IR  IREF  JREF  ICVHC  ICVHB
         1  1   1   0     0     CV120  CV120
         2  1   2   1     1
         3  3   ALL 1     1
         4  4   1   0     0     CV110  CV111
         5  4   2   4     1
         6  5-7 ALL 4     1
```

Core component masses must be input on the COR_CCM record, or in the COR_Kcc format, with cc corresponding to the component identifier (this is the only possible way of input for the new models and components).

COR_CCM – Cell Component Masses

Optional if a valid reference cell (IREF, JREF) is entered.

This record specifies the initial masses of the intact components in each cell. This record is not required if a valid reference cell (IREF, JREF) has been entered on input

record COR_RBV. Ten floating point fields in each table row (from fourth to thirteenth) are allowed on this record; however, any of these fields may be missing or set to -1.1 (any negative value may be used) if the value from the reference cell (IREF, JREF) is desired.

Equivalent information may optionally be provided, in a more general format, on input records COR_Kcc, where cc specifies a specific component.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMFU

Mass of UO_2 in the cell fuel component.

(type = real, default = -1.1, units = kg)

(5) XMCL

Mass of Zircaloy in the cell cladding component.

(type = real, default = -1.1, units = kg)

(6) XMCN

Total mass of Zircaloy in the two cell canister components. Unless specified otherwise in sensitivity coefficient array 1501, the total canister mass entered in this field will be divided equally between the two canister components (one component is adjacent to the control blade; the other is not). This field is ignored for PWR calculations (IRTYP = 'PWR') if field XMHT is not input.

(type = real, default = -1.1, units = kg)

(7) XMHT

Mass of electric heating element in the cell fuel component. If non-zero, a modified version of subroutine ELHEAT must be linked to MELCOR.

(type = real, default = -1.1, units = kg)

(8) XMIN

Mass of Inconel associated with the cell cladding component (e.g., PWR grid spacers).

(type = real, default = -1.1, units = kg)

(9) XMSSSS

Mass of steel in the cell supporting structure component.

(type = real, default = -1.1, units = kg)

(10) XMSSZR

Mass of Zircaloy in the cell supporting structure component.

(type = real, default = -1.1, units = kg)

(11) XMNSSS

Mass of steel in cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

(12) XMNSCP

Mass of control poison in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

(13) XMNSZR

Mass of Zircaloy in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

Instead of using record COR_CCM to input masses for the cell, optional input records for each component present in a cell may be used. This format is required for the PWR core shroud and for former components in MELCOR 1.8.6 and later. Any given record is not generally required, but at least one mass value for at least one component must be entered if this option is exercised. For a given cell, the two options may not be mixed; use of the COR_CCM record precludes use of any of the new-format component mass records. However, different options may be used for different cells.

Example

```
COR_CCM 2 ! IA IR FU  CL  CN  HT  IN  SSSS ZR  NSSS CP  ZR
          1 1  3  0.0 0.0 0.0 0.0 0.0 0.0  0.0 0.0  0.0 0.0
          2 2  3  0.0 0.0 0.0 0.0 0.0 0.0  0.0 0.0  0.0 0.0
```

COR_KFU – Fuel Component Masses

Optional.

This record specifies the initial masses of the fuel component in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the COR component. The table is arranged by elevation and ring indices making it easier tor read.

a. NUMSTR

Number of table records.

(type = integer, default = none, units = none)

b. CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials for the COR component (UO2, HEATING (heating element), ADD (additional fuel material), ADD_OXIDE(mass of oxide of additional fuel material)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

…

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! UO2 Masses in Fuel
COR_KFU 3 UO2  !IA  MUO2-1  MUO2-2  MUO2-3  MUO2-4  MUO2-5  : Ring 1-5
                 7  2038.0  2360.8  2975.6  3297.9  1311.0
                 8  2038.0  2360.8  2975.6  3297.9  1311.0
                 9  2038.0  2360.8  2975.6  3297.9  1311.0
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. One to four floating point fields in each row (from fourth to seventh) are allowed on this record. If any field on this record is missing or set to -1.1 or any negative value (or if the record is not present), the corresponding value from reference cell (IREF, JREF) is used if a valid cell (IREF, JREF) is defined; otherwise, the value is set to zero. For cases not involving electrical heating, the sixth field is permitted only if record COR_FUM is present to specify an additional material in FU. The seventh field is permitted only if that material is Zr or SS. For cases involving electrical heating, these fields specify the masses of the first and second electrode materials defined on record COR_RT.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMFUUO

Mass of UO_2 in the cell fuel component.

(type = real, default = -1.1, units = kg)

(5) XMFUHT

Mass of the electric heating element in the cell fuel component.

(type = real, default = -1.1, units = kg)

For calculations not involving electrical heating

(6) XMFUXM

Mass of additional fuel material (as defined on the COR_FUM record) in the cell fuel component.

(type = real, default = -1.1, units = kg)

(7) XMFUXO

Mass of oxide of additional fuel material (if it is Zr or SS) in the cell fuel component.

(type = real, default = -1.1, units = kg)

For cases involving electrical heating

(6) XMELM1

Mass of first electrode material in the cell fuel component.

(type = real, default = -1.1, units = kg)

(7) XMELM2

Mass of second electrode material in the cell fuel component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KFU 3 !IA  IR  XMFUUO    XMFUHT      XMFUXM     XMFUXO
        1  4   1  0.1481232   0.0
        2  5   1  0.2001665   0.0
        3  6   1  0.1801498   0.0
```

COR_KCL – Cladding Component Masses

Optional.

This record specifies the initial masses of the cladding component in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the clad component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the clad component (ZR, ZRO2, INC)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! ZRO2 Masses in Fuel
COR_KCL 3 ZRO2  !IA  MZRO2-1  MZRO2-2  MZRO2-3  MZRO2-4  MZRO2-5  : Ring 1-5
                  7   .550     .5631    .580      .782    .3002
                  8   .550     .5631    .580      .782    .3002
                  9   .550     .5631    .580      .782    .3002
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Three floating point fields in each table row (from fourth to sixth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMCLZR

Mass of Zircaloy in the cell cladding component.

(type = real, default = -1.1, units = kg)

(5) XMCLZX

Mass of ZrO_2 in the cell cladding component.

(type = real, default = -1.1, units = kg)

(6) XMCLIN

Mass of Inconel associated with the cell cladding component (e.g., PWR grid spacers).

(type = real, default = -1.1, units = kg)

Example

```
COR_KCL 3 !IA  IR   XMCLZR        XMCLZX        XMCLIN
       1   3   1   1.71919E-02      0.0          0.0
       2   4   1   3.63486E-02      0.0          0.0
       3   5   1   4.91197E-02      0.0          0.0
```

COR_KCN – Canister (CN) Component Masses

Optional, not permitted for SFP-PWR.

This record specifies the initial masses of the canister CN component (the portion not adjacent to a control blade) in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the CN component (ZR, ZRO2, INC)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! ZR Masses in CN
COR_KCN 3 ZR    !IA  MZR-1     MZR-2    MZR-3    MZR-4    MZR-5  : Ring 1-5
                  7  100.0     150.0    190.0    220.0    250.0
                  8  100.0     150.0    190.0    220.0    250.0
                  9  100.0     150.0    190.0    220.0    250.0
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Three floating point fields in each table row (from fourth to sixth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record

is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMCNZR

Mass of Zircaloy in the cell canister (CN) component.

(type = real, default = -1.1, units = kg)

(5) XMCNXI

Mass of ZrO_2 on the inside surface of the cell canister (CN) component.

(type = real, default = -1.1, units = kg)

(6) XMCNXO

Mass of ZrO_2 on the outside surface of the cell canister (CN) component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KCN  3 !  IA  IR   XMCNZR        XMCNXI        XMCNXO
          1   2   3    0.1357          0.0           0.0
          2   3   3    0.1348          0.0           0.0
          3   4   3    8.986E-02       0.0           0.0
```

COR_KCB – Canister (CB) Component Masses

Optional, not permitted for SFP-PWR.

This record specifies the initial masses of the canister CB component (the portion adjacent to a control blade) in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the CB component (ZR, ZRO2, INC)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! ZR Masses in CB
COR_KCB 3 ZR    !IA  MZR-1     MZR-2    MZR-3    MZR-4    MZR-5  : Ring 1-5
                  7  100.0     150.0    190.0    220.0    250.0
                  8  100.0     150.0    190.0    220.0    250.0
```

```
                    9  100.0     150.0    190.0    220.0    250.0
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Three floating point fields in each table row (from fourth to sixth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMCBZR

Mass of Zircaloy in the cell canister (CB) component.

(type = real, default = -1.1, units = kg)

(5) XMCBXI

Mass of ZrO_2 on the inside surface of the cell canister (CB) component.

(type = real, default = -1.1, units = kg)

(6) XMCBXO

Mass of ZrO_2 on the outside surface of the cell canister (CB) component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KCB  3 !IA  IR   XMCBZR        XMCBXI        XMCBXO
         1  7   1    477.810         0.0           0.0
         2  8   1    463.670         0.0           0.0
         3  9   1    463.670         0.0           0.0
```

COR_KSS – Supporting Structure Component Masses

Optional.

This record specifies the initial masses of the SS supporting structure component in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the SS component (SS,SSOX (or SSO2 or SSX), ZR, ZRO2 (or ZROX or ZX), GR)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer , default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

…

(NRING+1) Mass of material CMAT in ring NRING
(type = real, default = none, units = kg)

Example

```
! SS Oxide Masses in SS
COR_KSS 1 SSX    !IA  MSSX-1    MSSX-2   MSSX-3   MSSX-4   MSSX-5  : Ring 1-5
                    6   .005      .005     .005     .005     .005
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Four floating point fields in each table row (from fourth to seventh) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR
Number of table records.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Table record index.
(type = integer, default = none, units = none)

(2) IA
The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).
(type = integer / character*7, default = none, units = none)

(3) IR
The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).
(type = integer / character*7, default = none, units = none)

(4) XMSSSS
Mass of steel in the cell supporting structure component.
(type = real, default = -1.1, units = kg)

(5) XMSSSX
Mass of steel oxide in the cell supporting structure component.
(type = real, default = -1.1, units = kg)

(6) XMSSZR

Mass of Zircaloy in the cell supporting structure component.

(type = real, default = -1.1, units = kg)

(7) XMSSZX

Mass of ZrO_2 in the cell supporting structure component.

(type = real, default = -1.1, units = kg)

(8) XMSSCX

Mass of graphite in the cell supporting structure component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KSS 3 !IA IR  XMSSSS     XMSSSX     XMSSZR    XMSSZX      XMSSCX
       1   2  1   0.00619       0.        0.        0.         0.00619
       2   2  2   0.04949       0.        0.        0.         0.04949
       3   2  3   0.1003        0.        0.        0.         0.1003
```

COR_KNS – Nonsupporting Structure Component Masses

Optional, not permitted for SFP-BWR.

This record specifies the initial masses of the NS nonsupporting structure component in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the NS component (CRP (or CR for control poison), SS,SSOX (or SSO2 or SSX), ZR, ZRO2 (or ZROX or ZX))

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can

be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA
The axial level number.
(type = integer, default = none, units = none)

(2) Mass of material CMAT in ring 1
(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2
(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING
(type = real, default = none, units = kg)

Example

```
! CR Oxide Masses in NS
COR_KNS 3 CR    !IA  MCR-1    MCR-2   MCR-3   MCR-4   MCR-5  : Ring 1-5
                  7   8.2      11.2    13.5    12.3    11.8   5
                  8   8.2      11.2    13.5    12.3    11.85
                  9   8.2      11.2    13.5    12.3    11.8
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Four floating point fields in each table row (from fourth to eighth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR
Number of table records.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Table record index.
(type = integer, default = none, units = none)

(2) IA
The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMNSSS

Mass of steel in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

(5) XMNSSX

Mass of steel oxide in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

(6) XMNSCP

Mass of control poison in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

(7) XMNSZR

Mass of Zircaloy associated with the cell nonsupporting structure component (e.g., PWR control rod guide tubes).

(type = real, default = -1.1, units = kg)

(8) XMNSZX

Mass of ZrO_2 in the cell nonsupporting structure component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KNS 3 !IA IR  XMNSSS  XMNSSX  XMNSCP  XMNSZR  XMNSZX
        1  7  1   515.58    0.    35.4      0.     0.
        2  8  1   515.58    0.    35.4      0.     0.
        3  9  1   515.58    0.    35.4      0.     0.
```

COR_KSH – PWR Core Shroud Component Masses

Optional.

This record specifies the initial masses of the PWR core shroud component in cell (IA,IR). Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the SH component (SS,SSOX-IN (or SSO2-IN) for the inside surface, SSOX-OUT (or SSO2-OUT) for the outer surface)

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer / character*7, default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! SS Oxide Masses for inside surface of shroud
COR_KSH 3 SSOX-IN    !IA  MSSX-1    MSSX-2   MSSX-3   MSSX-4   MSSX-5  : Ring 1-5
                       8   .005      .005     .005     .005     .005 5
                       9   .005      .005     .005     .005     .0055
                      10   .005      .005     .005     .005     .005
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Three floating-point fields in each table row (from fourth to sixth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record

is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMSHSS

Mass of steel in the PWR shroud component.

(type = real, default = -1.1, units = kg)

(5) XMSHXI

Mass of steel oxide on the inside surface of the PWR shroud component.

(type = real, default = -1.1, units = kg)

(6) XMSHXO

Mass of steel oxide on the outside surface of the PWR shroud component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KSH  10 !IA  IR   XMSHSS        XMSHXI        XMSHXO
          1   8   5   950.600         0.0           0.0
          2   9   5   950.600         0.0           0.0
          3  10   5   950.600         0.0           0.0
```

COR_KFM – PWR Core Former Component Masses

Optional.

This record specifies the initial masses of the PWR core former component in cell (IA,IR). Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each material initially present in the CN component. The table is arranged by elevation and ring indices making it easier tor read.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) CMAT

Field indicating the material for which this table provides data. The following are recognized as available materials materials for the FM component (SS,SSOX (or SSO2), ZR, ZRO2 (or ZROX))

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial material masses in each ring associated with an axial elevation. Note that a hyphen can be used for any mass as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer , default = none, units = none)

(2) Mass of material CMAT in ring 1

(type = real, default = none, units = kg)

(3) Mass of material CMAT in ring 2

(type = real, default = none, units = kg)

...

(NRING+1) Mass of material CMAT in ring NRING

(type = real, default = none, units = kg)

Example

```
! FM Oxide Masses in FM
COR_KFM 3 SS    !IA  MSS-1    MSS-2   MSS-3   MSS-4   MSS-5  : Ring 1-5
                  7   8.2      11.2    13.5    12.3    11.8
                  8   8.2      11.2    13.5    12.3    11.8
                  9   8.2      11.2    13.5    12.3    11.8
```

Alternate format

This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Four floating-point fields in each table row (from fourth to seventh) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero. Note that steel is treated as the structural material.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMFMSS

Mass of steel in the PWR core former component.

(type = real, default = -1.1, units = kg)

(5) XMFMZR

Mass of Zircaloy associated with the PWR core former component.

(type = real, default = -1.1, units = kg)

(6) XMSHSX

Mass of steel oxide in the PWR core former component.

(type = real, default = -1.1, units = kg)

(7) XMSHZX

Mass of ZrO_2 in the PWR core former component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KFM  3  !IA  IR   XMFMSS   XMFMZR    XMSHSX    XMSHZX
         1   8   5    296.600    0.0       0.0       0.0
         2   9   5    296.600    0.0       0.0       0.0
         3  10   5    296.600    0.0       0.0       0.0
```

COR_KPD – Particulate Debris Component Masses

Optional.

This record specifies the initial masses of the particulate debris component in each cell. For a cell with a separate bypass region in a BWR, it is the mass in the channel region, and mass in the bypass region is entered on the COR_KPB record. This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Seven floating point fields in each table row (from fourth to tenth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMPDZR

Mass of Zircaloy in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(5) XMPDZX

Mass of ZrO_2 in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(6) XMPDUO

Mass of UO_2 in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(7) XMPDSS

Mass of steel in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(8) XMPDSX

Mass of steel oxide in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(9) XMPDCP

Mass of control poison in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

(10) XMPDIN

Mass of Inconel in the cell particulate debris component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KPD 3 !IA IR XMPDZR XMPDZX XMPDUO XMPDSS XMPDSX XMPDCP XMPDIN
        1  1  1   0.0    0.0    3.53    0.0    0.0   0.0    0.0
        2  2  1   0.0    0.0    10.95   0.0    0.0   0.0    0.0
        3  3  1   0.0    0.0    6.13    0.0    0.0   0.0    0.0
```

COR_KPB – Particulate Debris Component Masses in Bypass

Optional.

This record specifies the initial masses of the particulate debris component in the bypass of each cell. This record is not generally required, but component masses for at least one component must be entered if the new-format option is exercised for this cell. Seven floating point fields in each table row (from fourth to tenth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMPBZR

Mass of Zircaloy in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(5) XMPBZX

Mass of ZrO_2 in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(6) XMPBUO

Mass of UO_2 in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(7) XMPBSS

Mass of steel in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(8) XMPBSX

Mass of steel oxide in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(9) XMPBCP

Mass of control poison in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

(10) XMPBIN

Mass of Inconel in the cell bypass particulate debris component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KPB 3 !IA IR XMPBZR XMPBZX XMPBUO XMPBSS XMPBSX XMPBCP XMPBIN
        1  1  1   0.0    0.0    6.47    0.0    0.0   0.0    0.0
        2  2  1   0.0    0.0    1.83    0.0    0.0   0.0    0.0
        3  3  1   0.0    0.0    7.14    0.0    0.0   0.0    0.0
```

COR_KRK – SFP Rack (RK) Component Masses

Optional, permitted only for SFP-BWR and SFP-PWR.

This record specifies the initial masses of the SFP Rack structure component in each cell. Five floating point fields in each table row (from fourth to eighth) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMRKSS

Mass of steel in the cell SFP Rack structure component.

(type = real, default = -1.1, units = kg)

(5) XMRKSX

Mass of steel oxide in the cell SFP Rack structure component.

(type = real, default = -1.1, units = kg)

(6) XMRKCP

Mass of control poison in the cell SFP Rack structure component.

(type = real, default = -1.1, units = kg)

(7) XMRKZR

Mass of Zircaloy associated with the cell SFP Rack structure component (e.g., PWR control rod guide tubes).

(type = real, default = -1.1, units = kg)

(8) XMRKZX

Mass of ZrO_2 in the cell SFP Rack structure component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KRK 3 !IA IR XMRKSS XMRKSX XMRKCP XMRKZR XMRKZX
        1  1  1   1.2    0.0    0.00    0.0    6.7
        2  2  1   0.0    6.4    0.76    0.0    1.5
        3  3  1   0.0    0.0    5.14    0.0    0.0
```

COR_KRF – Reflector Component Masses

Optional, permitted only for PBR and PMR.

This record specifies the initial masses of the reflector components in each cell. One floating point field in each table row is allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) XMRF

Mass of graphite in the cell reflector component.

(type = real, default = -1.1, units = kg)

Example

```
COR_KRF 1 !IA    IR  XMRF
        1  5-21  6   4400.0
```

COR_CDM – Conglomerate Debris Masses

Optional.

This record specifies the initial conglomerate debris masses for component IC in each cell. This record is not required. Seven floating point fields in each table row (from fifth to eleventh) are allowed on this record. If any fields on this record are missing or set to -1.1 or any negative value (or if the record is not present), the corresponding values from reference cell (IREF, JREF) are used if a valid cell (IREF, JREF) is defined; otherwise, the values are set to zero.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) IC

Component identifier. Permitted components are

(a) 2
CL

(b) 3
CN

(c) 4
CB

(d) 6
PD

(e) 7
SS

(f) 8
NS

(g) 9
PB

(type = integer / character*2, default = none, units =none)

(5) YMCCZR
Mass of Zircaloy in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(6) YMCCZX
Mass of ZrO_2 in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(7) YMCCUO
Mass of UO_2 in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(8) YMCCSS
Mass of steel in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(9) YMCCSX
Mass of steel oxide in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(10) YMCCCP
Mass of control poison in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

(11) YMCCIN
Mass of Inconel in the conglomerate debris for component IC.
(type = real, default = -1.1, units = kg)

Example

```
COR_CDM 3 !A R IC YMCCZR YMCCZX YMCCUO YMCCSS YMCCSX YMCCCP YMCCIN
        1  1 1 CL  1.2    0.0    0.00    0.0    6.7    0.0    0.0
        2  2 1 SS  0.0    6.4    0.76    0.0    1.5    0.0    0.0
        3  3 1 PB  0.0    0.0    5.14    0.0    0.0    0.0    0.0
```

<u>Core Component Temperatures</u>

COR_CIT – Initial Component Temperatures

This record specifies the initial component temperatures for each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse. Required

More readable form

A table is provided for each COR component. The table is arranged by elevation and ring indices making it easier tor read. Using this format deprecates the need for the COR_PCT, and COR_SRT records.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) Component

Field indicating the component for which this table provides data. The following are recognized as available components (FU, CL, CN, CB, SH, FM, PD, SS, NS, PB, MP1, MP2, MB1, MB2, RK, RF)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides initial temperatures for each ring associated with an axial elevation. Note that a hyphen can be used for any temperature as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Temperature for ring 1

(type = real, default = none, units = K)

(3) Temperature for ring 2

(type = real, default = none, units = K)

…

(NRING+1) Temperature for ring NRING

(type = real, default = none, units = K)

Example

```
! CLAD Temperatures
COR_CIT 3  CL!IA  TCL1   TCL2   TCL3   TCL4   TCL5 : Ring 1-5
               8   555    553    551    550    545
```

```
  9   558    556    553    551    545
 10   560    558    555    552    546
```

Alternate form
Optional if a valid reference cell (IREF, JREF) is entered.

This record specifies the initial component temperatures for each cell. This record is not required if a valid cell (IREF, JREF) has been entered. Eight floating point fields in each table row (from fourth to eleventh) are allowed on this record; however, any of these fields may be missing or set to -1.1 or any negative value if the value from a reference cell (IREF, JREF) is desired.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) TFU

UO_2 fuel temperature.

(type = real, default = -1.1, units = K)

(5) TCL

Cladding temperature.

(type = real, default = -1.1, units = K)

(6) TCN

Canister (CN) temperature. This field is ignored for PWR calculations (IRTYP = PWR). Not used for SFP-PWR.

(type = real, default = -1.1, units = K)

(7) TCB

Canister (CB) temperature. If omitted, the CB component will be initialized to the CN component temperature entered on the field TCN. This field is ignored for PWR calculations (IRTYP = PWR). Not used for SFP-PWR.

(type = real, default = -1.1, units = K)

(8) TPD

Particulate debris temperature. If omitted, the value from reference cell (IREF, JREF) is used if a valid cell (IREF,JREF) is defined; otherwise, the value is set to zero.

(type = real, default = -1.1, units = K)

(9) TSS

Supporting structure temperature.

(type = real, default = -1.1, units = K)

(10) TNS

Nonsupporting structure temperature.

(type = real, default = -1.1, units = K)

(11) TPB

Particulate debris temperature in the bypass. If omitted, the value from reference cell (IREF, JREF) is used if a valid cell (IREF, JREF) is defined; otherwise, the value is set to zero.

(type = real, default = -1.1, units = K)

Example

```
COR_CIT 3 !IA IR  TFU  TCL  TCN  TCB  TDP  TSS   TNS  TDB
        1  1  1   0.0  0.0  0.0  0.0  0.0  551.0 0.0  0.0
        2  2  1   0.0  0.0  0.0  0.0  0.0  551.0 0.0  0.0
        3  3  1   0.0  0.0  0.0  0.0  0.0  551.0 0.0  0.0
```

COR_PCT –Initial PWR Component Temperatures

Optional if a valid reference cell (IREF, JREF) is entered.

This record specifies the initial component of the PWR core shroud and former temperatures for cell (IA,IR) and is used only for a PWR (IRTYPE=PWR). Two floating-point fields in each table row (fourth and fifth) are allowed on this record; however, either of these fields may be missing or set to -1.1 or any negative value if the value from a reference cell (IREF, JREF) is desired.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) TSH

Core shroud (baffle) temperature.

(type = real, default = -1.1, units = K)

(5) TFM

Core former temperature.

(type = real, default = -1.1, units = K)

Example

```
COR_PCT   3 !IA  IR    TSH         TFM
          1   9   5   568.0       562.0
          2  10   5   568.0       562.0
          3  11   5   568.0       562.0
```

COR_SRT – Cell Initial SFP-Rack Temperature

Optional if a valid reference cell (IREF, JREF) is entered,
processed only for SFP-BWR and SFP-PWR.

This record specifies the initial SFP Rack temperature for each cell. This record is not required if a valid cell for (IREF, JREF) has been entered. One floating point field in each table row is allowed on this record; however, it may be missing or set to -1.1 or any negative value if the value from the reference cell (IREF, JREF) is desired.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) TRK

SFP Rack temperature.

(type = real, default = -1.1, units = K)

Example

```
COR_SRT   3 !IA  IR    TRK
          1   9   5   500.0
          2  10   5   500.0
          3  11   5   500.0
```

Core Component Equivalent Diameters

COR_EDR – Equivalent Diameter Record

Required

This record specifies the equivalent diameter (for convective heat transfer) for each component in each cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each component present in the problem. The table is arranged by elevation and ring indices making it easier tor read. The header record provides the number of records in the table and specifies the component associated with the table. Using this format deprecates the need for the COR_CDR, COR_SDR, COR_RDR, and COR_RFD records.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) COMP

Field indicating the component for which this table provides data. The following are recognized as available components (FU, CL, CN, CB, SH, FM, PD, SS, NS, PB, MP1, MP2, MB1, MB2, RK, RF)

For the CN component, the user can specify the inside or channel (CNC) equivalent diameter and the outside or bypass (CNB) to be used for both CN and CB components. Alternatively, the user can specify both the bypass and channel surface areas for both CN and CB components independently, i.e., CN-CH, CN-BP, CB-CH, and CB-BP. This specification deprecates the need for the COR_CDR record.

For the SH components, the user can specify the equivalent diameter associated with bypass and channel separately, i.e., SH-CH, SH-BP. This specification deprecates the need for the COR_SDR record.

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides equivalent diameters in each ring associated with an axial elevation for the component specified on the table header record. Note that a hyphen can be used for any equivalent diameter as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Equivalent diameter of component COMP in ring 1

(type = real, default = none, units = m)

(3) Equivalent diameter of component COMP in ring 2

(type = real, default = none, units = m)

...

(NRING+1) Equivalent diameter of component COMP in ring NRING

(type = real, default = none, units = m)

Example

```
! Equivalent diameter associated with CN component
COR_EDR 3 CN    !IA  EDR-1    EDR-2   EDR-3   EDR-4   EDR-5  : Ring 1-5
                  7   .043     .043    .043    .043    .043
                  8   .043     .043    .043    .043    .043
                  9   .043     .043    .043    .043    .043
```

Alternate format

This record is not required if a valid cell (IREF, JREF) has been entered. Seven floating point fields in each table row (from fourth to tenth) are allowed on this record; however, any of these fields may be missing or set to -1.1 or any negative value if the value from the reference cell (IREF, JREF) is desired.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DHYCL

Cladding equivalent outside diameter.

(type = real, default = -1.1, units = m)

(5) DHYPD

Particulate debris equivalent diameter. Because it is used in calculating total debris surface area, this diameter is of particular importance in calculating quenching heat transfer during material relocation from the core to the lower plenum in the falling debris quench model. If desired, a different value can be specified as DHYPB for debris in the bypass of a BWR.

(type = real, default = -1.1, units = m)

(6) DHYCNC

Canister inside equivalent diameter. The same value is used for both canister components. This field is ignored for PWR calculations (IRTYP = PWR). Not used for SFP-PWR.

(type = real, default = -1.1, units = m)

(7) DHYCNB

Canister outside equivalent diameter. Currently, the same value is used for both canister components (i.e., the fact, that one canister component is adjacent to the control blade and the other is not, is not taken into account). This field is ignored for PWR calculations (IRTYP = PWR). Not used for SFP-PWR

(type = real, default = -1.1, units = m)

(8) DHYSS

Supporting structure equivalent diameter.

(type = real, default = -1.1, units = m)

(9) DHYNS

Nonsupporting structure equivalent diameter. Not used for SFP-BWR.

(type = real, default = -1.1, units = m)

(10) DHYPB

Particulate debris equivalent diameter in the bypass of a BWR. If this field is omitted, the diameter of particulate debris in the channel will be used.

(type = real, default = DHYPD, units = m)

Example

```
COR_EDR 3 !IA IR DHYCL   DHYPD  DHYCNC DHYCNB  DHYSS DHYNS DHYPB
        1  1  1 1.22E-02 2.5E-02 0.132 2.2E-02 1.195 1.195 2.5E-02
        2  2  1 1.22E-02 2.5E-02 0.132 2.2E-02 0.154 0.154 2.5E-02
  3  3  1 1.22E-02 2.5E-02 0.132 2.2E-02 0.154 0.154 2.5E-02
```

COR_CDR – Canister Equivalent Diameter Record

Optional, processed only for BWR and SFP-BWR.

This record specifies the equivalent diameter (for convective heat transfer) for surfaces of the canister and canister-b components in cell (IA,IR) and is used only for a BWR (IRTYP=BWR). If it is absent, the values input on record COR_EDR, which do not distinguish between the two canister components, CN and CB, will be used.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all seven fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DHYCNC

Channel-side canister equivalent diameter.

(type = real, default = none, units = m)

(5) DHYCNB

Bypass-side canister equivalent diameter.

(type = real, default = none, units = m)

(6) DHYCBC

Channel-side canister-b equivalent diameter.

(type = real, default = none, units = m)

(7) DHYCBB

Bypass-side canister-b equivalent diameter.

(type = real, default = none, units = m)

Example

```
COR_CDR 3 ! IA IR   DHYCNC   DHYCNB   DHYCBC   DHYCBB
        1   1  1     1.5      2.5      1.32     2.2
        2   2  1     1.6      2.5      1.32     2.2
        3   3  1     1.7      2.5      1.32     2.2
```

COR_SDR – PWR Shroud and Former Equivalent Diameter Record

Optional.

This record specifies the equivalent diameter (for convective heat transfer) for surfaces of the PWR shroud and former components in cell (IA,IR) and is used only for a PWR (IRTYP=PWR).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all six fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DHYSHC

Channel-side core shroud equivalent diameter.

(type = real, default = none, units = m)

(5) DHYSHB

Bypass-side core shroud equivalent diameter.

(type = real, default = none, units = m)

(6) DHYCBC

Core former equivalent diameter.

(type = real, default = none, units = m)

Example

```
COR_SDR 3 ! IA  IR   DHYSHC   DHYSHB   DHYCBC
        1   1   1     2.5      2.5      3.32
        2   2   1     2.6      2.5      1.32
        3   3   1     2.7      2.5      5.32
```

COR_RDR – SFP Rack Equivalent Diameter Record

Optional, processed only for SFP-BWR and SFP-PWR.

This record specifies the equivalent diameters (for convective heat transfer) for surfaces of the SFP rack in each cell.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all four fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DHYRK

SFP Rack equivalent diameter.

(type = real, default = none, units = m)

Example

```
COR_RDR 3 ! IA  IR   DHYRK
        1   1   1     3.5
        2   2   1     2.6
  3   3   1     4.7
```

COR_RFD – Reflector Equivalent Diameters Record

Optional, processed only for PBR and PMR.

This record specifies the inner and outer (surface 1 and 2) reflector equivalent hydraulic diameters to be used for heat transfer coefficients. Note that these are not necessarily the same as actual reflector diameters.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all five fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) DHYRFC

Reflector inner equivalent diameter.

(type = real, default = none, units = m)

(5) DHYRFB

Reflector outer equivalent diameter.

(type = real, default = none, units = m)

Example

```
COR_RFD 1 ! IA   IR   DHYRFC  DHYRFB
        1   5-10 6    0.06    0.
```

COR_RFG –Reflector Geometry Record

Optional, processed only for PBR and PMR.

This record specifies the reflector geometry parameters.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all five fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a*–*b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) RADI

Reflector channel-side radius. For a flat top/bottom reflector, the channel side is the side towards the fueled region.

(type = real, default = none, units = m)

(5) THKRF

The reflector thickness. The sign is used to reference where the bypass side is relative to the channel side, so for an outer reflector, this would be positive and the outer side is the bypass side; for an inner reflector, the channel side is the outer side, so this is negative to indicate that the bypass side is inside. Similarly, a top reflector has the channel side down, so THKRF is positive, and the bottom reflector has the channel side on the upper side, so THKRF is negative.

(type = real, default = none, units = m)

(6) IGEOMRF

The geometry flag

(a) 0
Flat plate

(b) 1
Cylindrical

(type = integer, default = 1, units = none)

(7) FACRF

The factor to be used to split the thermal conductances to the inner and outer surfaces. If not entered, it is calculated internally.

(type = real, default = calculated, units = none)

Example

```
COR_RFG 1 ! IA   IR   RADI  THKRF  IGEOMRF
        1   5-10 6    2.5   1.0    1
```

COR_BFA – Cell Boundary and Flow Areas Record

Required

This record specifies the outer radial boundary area and the intact cross-sectional flow areas for each cell. Two formats are available. The first format presented is more

readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each component present in the problem. The table is arranged by elevation and ring indices making it easier tor read. The header record provides the number of records in the table and specifies the component associated with the table.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(2) AREAX

Field indicating the type of area specified in this table. The following are recognized as available area types ASCELR (Radial cell boundary), AFLOWC (Channel flow area), and AFLOWB (Bypass flow area)

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides equivalent diameters in each ring associated with an axial elevation for the component specified on the table header record. Note that a hyphen can be used for any area as a placeholder if that cell is described from a valid reference cell.

(1) IA

The axial level number.

(type = integer, default = none, units = none)

(2) Area of type AREAX in ring 1

(type = real, default = none, units = m^2)

(3) Area of type AREAX in ring 2

(type = real, default = none, units = m^2)

...

(NRING+1) Area of type AREAX in ring NRING

(type = real, default = none, units = m^2)

Alternate format

The flow areas will be automatically reduced to account for any debris masses present in the cell. This record is not required if a valid reference cell (IREF,JREF) has been entered. Three floating point fields (from fourth to sixth) are allowed on this record; however, any of these fields may be missing or set to -1.1 if the value from the reference cell (IREF, JREF) is desired.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(2) NSTR

Table record index.

(type = integer, default = none, units = none)

(3) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(5) ASCELR

Area of outer radial cell boundary ($2\pi R \cdot DZ$, where R is the outer radius of ring IR and DZ is the axial level length of level IA).

(type = real, default = -1.1, units = m^2)

(6) AFLOWC

Channel flow area of cell.

(type = real, default = -1.1, units = m^2)

(7) AFLOWB

Bypass flow area of cell in a BWR. For a PWR (IRTYP = 'PWR'), AFLOWB will simply be added to AFLOWC except with cells in the barrel/baffle region (if IOLDOS = 0).

(type = real, default = -1.1, units = m^2)

Example

```
COR_BFA 3 !IA  IR   ASCELR    AFLOWC    AFLOWB
      1   1   1   1.492     1.14642     0.0
      2   2   1   1.492     1.14642     0.0
      3   3   1   1.492     1.14642     0.0
```

<u>Core Component Surface Areas</u>

COR_SA – Surface Area Record

Required

This record specifies the intact surface area for each intact component in each cell. These areas will be automatically adjusted to account for any debris masses present in the cell. Two formats are available. The first format presented is more readable as it appers in spreadsheet fashion. The alternative format is less easily interpreted but may be more concise if input is sparse.

More readable format

A table is provided for each component present in the problem. The table is arranged by elevation and ring indices making it easier tor read. The header record provides the number of records in the table and specifies the component associated with the table. Using this format deprecates the need for the COR_CSA, COR_SSA, COR_RSA, and COR_RFA records.

(3) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(4) COMP-S

Field indicating the component surface for which this table provides data. The following are recognized as available components (FU, CL, CN, SH, FM, SS, NS, RK).

For the CN component, the user can specify the total surface area associated with both CN and CB components, which will be divided equally between the two canister components, unless the fractional partition is specified otherwise in sensitivity coefficient array 1501. Alternatively, the user can specify both the bypass and channel surface areas for both CN and CB components independently, i.e., CN-CH, CN-BP, CB-CH, and CB-BP. This specification deprecates the need for the COR_CSA record.

For the SH and FM components, the user can specify the surface area associated with bypass and channel separately, i.e., SH-CH, SH-BP, FM-CH, FM-BP. This specification deprecates the need for the COR_SSA record.

(type = character*7, default = none, units = none)

The following data are input as a table with length NUMSTR. The table provides equivalent diameters in each ring associated with an axial elevation for the component specified on the table header record. Note that a hyphen can be used for any area as a placeholder if that cell is described from a valid reference cell.

(4) IA

The axial level number.

(type = integer, default = none, units = none)

(5) Surface area of COMP-S in ring 1
(type = real, default = none, units = m^2)

(6) Surface area of COMP-S in ring 2
(type = real, default = none, units = m^2)

...

(NRING+1) Surface area of COMP-S in ring NRING
(type = real, default = none, units = m^2)

Example

```
! Surface area associated with SH component
COR_SA 3 SH-CH   !IA  SHC-1    SHC-2   SHC-3   SHC-4   SHC-5  : Ring 1-5
                  7     -        -       -      .043    -
                  8     -        -       -      .043    -
                  7     -        -       -      .043    -
COR_SA 3 SH-BP   !IA  SHB-1    SHB-2   SHB-3   SHB-4   SHB-5  : Ring 1-5
                  7     -        -       -      .043    -
                  8     -        -       -      .043    -
                  7     -        -       -      .043    -
```

Alternate format

This record is not required if a valid reference cell (IREF,JREF) has been entered. Five floating point fields in each table row (from fourth to eighth) are allowed on this record; however, any of these fields may be missing or set to -1.1 if the value from the reference cell (IREF,JREF) is desired.

(1) NUMSTR
Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Table record index.

(type = integer, default = none, units = none)

(2) IA
The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ASFU

Fuel surface area.

(type = real, default = -1.1, units = m^2)

(5) ASCL

Cladding surface area.

(type = real, default = -1.1, units = m^2)

(6) ASCN

Total canister inside surface area. The value entered in this field will be divided equally between the two canister components, unless specified otherwise in sensitivity coefficient array 1501 or overridden by input on a COR_CSA CORijj06A record. This field is ignored for PWR calculations (IRTYP = PWR).

(type = real, default = -1.1, units = m^2)

(7) ASSS

Supporting structure surface area.

(type = real, default = -1.1, units = m^2)

(8) ASNS

Nonsupporting structure surface area.

(type = real, default = -1.1, units = m^2)

Example

```
COR_SA 3 !IA  IR  ASFU  ASCL  ASCN  ASSS  ASNS
       1  1   1   0.0   0.0   0.0   121.08 0.0
       2  2   1   0.0   0.0   0.0   24.33  0.0
       3  3   1   0.0   0.0   0.0   24.33  0.0
```

COR_CSA – Canister Surface Area Record

Optional.

This record specifies the intact surface area for surfaces of the canister and canister-b components in cell (IA, IR) and is used only for a BWR (IRTYP=BWR). If it is absent, these areas will be derived from the value of ASCN on record COR_SA and from partitioning coefficients in sensitivity coefficient array 1501 or, if that record is absent, inherited from the reference cell, IREFN, specified on record COR_RBV.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all seven fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ASCN

Channel-side canister surface area.

(type = real, default = none, units = m^2)

(5) ASCNB

Bypass-side canister surface area.

(type = real, default = none, units = m^2)

(6) ASCB

Channel-side canister-b surface area.

(type = real, default = none, units = m^2)

(7) ASCBB

Bypass-side canister-b surface area.

(type = real, default = none, units = m^2)

Example

```
COR_CSA 3 ! IA  IR  ASCN  ASCNB  ASCB  ASCBB
         1  1   1   0.0   0.0    0.0   21.08
         2  2   1   0.0   0.0    0.0   34.33
         3  3   1   0.0   0.0    0.0   54.33
```

COR_SSA – PWR Shroud and Former Surface Area Record

Optional.

This record specifies the intact surface area for surfaces of the PWR shroud and for former components in cell (IA, IR) and is used only for a PWR (IRTYP).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all six fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ASSHC

Channel-side PWR core shroud surface area.

(type = real, default = none, units = m^2)

(5) ASSHB

Bypass-side PWR core shroud surface area.

(type = real, default = none, units = m^2)

(6) ASFM

PWR core former surface area.

(type = real, default = none, units = m^2)

Example

```
COR_SSA 3 ! IA  IR   ASSHC        ASSHB     ASFM
         1   3  3  2.588E-02    2.588E-02  1.E-08
         2   4  3  2.588E-02    2.588E-02  0.0
         3  15  3  3.209E-02    3.209E-02  0.0
```

COR_RSA – SFP Rack Surface Area Record

Optional,

processed only for SFP-BWR and SFP-BWR.

This record specifies the total surface area (both sides) of the SFP Rack in each cell.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all four fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* ([without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ASRK

SFP Rack surface area.

(type = real, default = none, units = m^2)

Example

```
COR_RSA 1 ! IA  IR   ASRK
         1   3  3  2.588E-02
```

COR_RFA – Reflector Surface Area Record

Optional,
processed only for PBR and PMR.

This record specifies the surface of the reflector in each cell.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR. If the record is present, all five fields must be present in each row.

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* ([without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ASRF

Channel-side reflector surface area.

(type = real, default = none, units = m^2)

(5) ASRFB

Bypass-side reflector surface area.

(type = real, default = none, units = m^2)

Example

```
COR_RFA 1 ! IA  IR   ASRF    ASRFB
         1   3  3  2.588E-02 3.6E-02
```

COR_FCEL – Local Radiative Exchange Factor

Optional.

FCELR and FCELA are defined globally on the COR_RF record. However, the user can specify a local value for FCELR and FCELA on the table that follows this header record. This local value can be specified as a constant, a local control function, or, in the case of FCELR, the user can request it be calculated by an internal model (described in the COR reference manual).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

(1) FCELCtype

User indicates whether the following table provides data for either the 'FCELR' (radial radiation exchange) or 'FCELA' (axial radiation exchange) parameter.

(type = character*7, default = none, units = none)

The following table of data with length NUMSTR is provided:

(1) IA

The axial level number

(type = integer, default = none, units = none)

(2 to NRAD+1) FCELx (IR)

NRAD fields are available to specify the local value for FCELx (either FCELR or FCELA depend on FCELtype specified on the header card). If the user chooses to omit an input for a particular ring (using the global value), a hyphen '-' is used as a placeholder in the table. The type of each field indicates the meaning of the value that is provided. The user can specify either a local constant, a control function, or the internal model. A real number indicates a local constant, an integer indicates a control function number and a character indicates a control function name. However, if the character field is the word 'MODEL', then this indicates that the local model is used. This is summarized as follows, dependent on the variable type:

(c) Real
Constant value

(d) Integer
Control function number

(e) Character*7 (except the word 'MODEL')
Control function name

(f) 'MODEL'
Use the local model described in the reference manual.
Currently there is no model available for FCELA.

(type = integer / real/ character*7, default = global value on COR_RF, units = none)

Example

```
COR_FCEL  5  FCELR
    7    .5      -       0.0   !IA=7,  FCELR(IR=1), FCELR(IR=2), FCELR(IR=3)
    8    .5   0.25    0.0
    9    .5   0.25    0.0
    10    MODEL   MODEL   0.0
    11   -   5   -      !CF-5 used for ring 2(alternatively could specify CF character name)
```

COR_NOX – Oxidation Shutoff Control Function Record

Optional.

This record specifies logical control functions to control various models on a cell-by-cell basis. A value of 0 / NO for ICFNOX indicates that no control function is to be used. If this record is absent, values default to those for reference cell (IREF, JREF), if it is defined, or to 0; a value of -1 / REF indicates that the value from the reference cell is to be used.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ICFNOX

Oxidation shutoff control function key.

(a) 0 or NO

No control function is to be used.

(b) 1 or CF

Use oxidation shutoff control function.

(c) -1 or REF

The value from the reference cell is to be used.

(type = integer / character*3, default = NO or REF, units = none)

If ICFNOX = CF then the following field is required:

(5) LCFNAM

Oxidation shutoff control function. A control function value of .TRUE. will shut off oxidation in the given cell (IA, IR) (e.g., to model the

effects of a flow blockage) for all surfaces connected to the channel control volume (ICVHC on record COR_RBV CORijj01).

(type = character*16, default = none, units = none)

Example

```
COR_NOX 2 !IA  IR  ICFNOX  LCFNAM
        1  1   2   CF      CF205
        2  5   1   REF
```

COR_CFF – Control Function Component Failure

Optional.

This record specifies a LOGICAL control function that will enforce the collapse of component ICOMP (forth field in each table row) in cell (IA,IR). If the function value becomes .TRUE. at a time when fuel rods are still standing in cell (IA,IR), they will be immediately converted to rubble (PD), independent of any other conditions in the cell.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IA

The axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(3) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) ICOMP

The component identification:

(a) CL
Cladding

(b) CN
Canister

(c) CB
Canister adjacent to blade

(type = character*2, default = none, units = none)

(5) ICFAI

User name of a LOGICAL control function that enforces the collapse of the specified component in core cell (IA,IR).

(type = character*16, default = none, units = none)

Example

```
COR_CFF   3 !IA  IR COMPONENT ICFAI
          1   7   1     CL    CF2007
          2   8   1     CL    CF2017
          3   9   1     CL    CF2027
```

COR_QHS – Control Function Heat Source

Optional.

In some calculations, it may be desirable to consider a heat source in a core component that does not correspond to fission, decay, or electrical heating. This record allows the specification of user-defined heat sources in COR components. ICFQ specifies the user name of a REAL control function whose value will be interpreted as a heat source (in W) to be added to component cc in core cell (IA,IR).

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(2) NSTR

Table record index.

(type = integer, default = none, units = none)

(3) IA

The 105axial level number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(4) IR

The radial ring number or the range of numbers (input as *a–b* [without any blanks] or the character string ALL).

(type = integer / character*7, default = none, units = none)

(5) ICOMP

The component identification:

(a) FU
Intact fuel pellets

(b) CL
Cladding

(c) CN
Canister

(d) CB
Canister adjacent to blade

(e) SH
Intact PWR core shroud (baffle)

(f) FM
Intact PWR core formers

(g) PD
Particulate debris

(h) SS
Core support structure

(i) NS
Nonsupporting other structures

(j) PB
Particulate debris in bypass (if any)

(k) MP1
Oxidic molten pool (portion in the channel)

(l) MB1
Oxidic molten pool in the bypass (if any)

(m) MP2
Metallic molten pool (portion in the channel)

(n) MB2
Metallic molten pool in the bypass (if any)

(type = character*3, default = none, units = none)

(6) ICFQ

User name of a REAL control function whose value specifies an additional heat source (in W) to be added to component ICOMP in core cell (IA,IR).

(type = character*16, default = none, units = none)

Example

```
COR_QHS 3 !  IA IR IC ICFQ
         1   1  1  PD CF101
         2   2  1  PD CF102
         3   3  1  PD CF103
```

2.5 MELGEN Lower Head Input

A major change from MELCOR 1.8.5 is that there is no longer a one-to-one correspondence between segments of the lower head and the rings of the core nodalization. The input must now contain an input record for each lower head segment.

The first NLHTA input records specify the outer radii of those segments of the lower head that are not associated with the cylindrical wall of the vessel. The list must include all of the ring radii input as RINGR on record COR_RP. If that portion of the head is curved (ILHTYP = 0 on record COR_VP), the list must also include the radius of any, and all, intersections of the curved portion of the lower head with axial elevations input as Z on the COR_ZP record. Such radii must be calculated from other input, and MELGEN requires that the match be very close (to six figures). However, if approximate values are input, it will compute and report precise values as part of the error message generated.

The model also includes a lower head segment for each axial level associated with the cylindrical portion of the head, for a total of NLHT ≥ NLHTA segments, and there must also be an input record for each such segment. MELGEN will issue a diagnostic message if the total number of records found is incorrect. If the bottom of the head is curved, it is often convenient to begin with input for NLHTA segments and allow MELGEN to determine how many more must be added.

MELCOR 1.8.6 no longer assumes that the outer surface of the entire lower head communicates with a single CVH control volume. Therefore, the appropriate volume must be specified for each head segment.

COR_LP – In-Vessel Falling Debris Quench Model Parameters

Required.

This record specifies parameters needed to activate the in-vessel falling debris quench model. This record is required; the first field must be present, while all other fields are optional. For more information on how these parameters are used in the model, see the COR Package Reference Manual, Section 2.3.7.

(1) IAXSUP

Axial level number of core cells containing the core support plate. It is used primarily as the starting point for tracking falling debris.

(type = integer, default = none, units = none)

(2) HDBH2O

Heat transfer coefficient from in-vessel falling debris to pool. A value of zero turns the model off. User Note: Large values of this parameter could result in vessel over-pressurization.

(type = real, default = 100.0, units = W/m^2-K)

(3) PPFAIL

Differential pressure between lower plenum volume and reactor cavity volume that will fail the lower head. This value must not exceed the critical pressure of water (2.2e7 Pa) or a fatal input error will result.

(type = real, default = 2.0e7, units = Pa)

(4) VFALL

Velocity of falling debris.

(type = real, default = 1.0 [In previous versions of MELCOR this value was defaulted to 5 m/s. The current default reflects greater resistance to falling offered by the water and rising bubbles.], units = m/s)

(5) NTLP

Number of axial segments in the lower plenum.

(type = integer, default = IAXSUP, units = none)

Example

```
!       IAXSUP  HDBH2O   PPFAIL    VFALL   NTLP
COR_LP  1       50.0     21800000. 0.5       1
```

COR_LH – Lower Head Structure Record

Required.

This record specifies the lower plenum nodalization parameters and is required. Note that the number of material layers in the lower head will be one less than the number of temperature nodes (NLH-1) because the temperature nodes are defined at the layer boundaries.

(1) NLH

Number of temperature nodes in the vessel lower head (minimum = 2).

(type = integer, default = none, units = none)

(2) NINSLH

Total number of insulation (nonload bearing) mesh layers in the vessel lower head (maximum = NLH-1).

Optional field.

(type = integer, default = 0, units = none)

Example

```
COR_LH   20   0
```

COR_LHD – Lower Head Segments Record

Required.

Each data line specifies information for one segment of the lower head. NLHTA of them will be associated with the noncylindrical portion of the lower head. The remaining NLHT-NLHTA will be associated with the cylindrical portion, one for the segment in each axial level of the nodalization.

Note that NLHT is a required parameter. If the user does not wish to precalculate it, we suggest starting with NLHTA records and obtaining the required total number from the diagnostic message that will be issued by MELGEN.

The input variables (6)-(10) are optional. If the heat structure name IHSCAV is present, then radiation from the lower head segment to the heat structure is enabled. The following three variables used by the radiation model are optional. If (8) or (9) are entered, then the emissivity calculated by the specified control function will be used instead of the internally calculated one (HS package default emissivity for steel). The last parameter, ICFCNV, specifies a control function that overrides the internal heat transfer coefficient calculated for the cavity pool

(1) NLHT

Number of lower head segments

(type = integer, default = none, units = none)

(2) NLHTA

Number of segments in noncylindrical (flat or hemispherical) portion of the lower head.

(type = integer, default = none, units = none)

The following data are input as a table with length NLHT:

(1) IS

Segment number.

(type = integer, default = none, units = none)

(1) TLH

Initial temperature of lower head nodes in head segment IS.

(type = real, default = none, units = K)

(2) RADLH

Outer radius of head segment IS. RADLH must equal RVESS for IS = NLHTA; the input value is not used for IS > NLHTA. As a convenience, the user may specify a particular outer ring radius by entering Ri, where i is the ring radius.

(type = real, default = none, units = m)

(3) ICVCAV

Reactor cavity control volume name. This field is required on all COR_LHD record lines.

(type = character*16, default = none, units = none)

(4) IHSCAV

Reactor cavity heat structure name for radiation from outer surface of lower head. Optional.

(type = character*16, default = none, units = none)

If IHSCAV is entered, (6) must be entered. (7) is optional, but the default view factor is 0.0 so the user will probably want to enter this.

(5) HSSIDE

Left or right side of heat structure.

(a) LHS
Left side

(b) RHS
Right side

(type = character*16, default = none, units = none)

(6) VFLHHS

View Factor from outer surface to heat structure. Optional.

(type = real, default = 0.0, units = none)

(7) ICFELH

Control function name for emissivity of outer surface. Optional.

(type = character*16, default = none, units = none)

(8) ICFEHS

Control function name for emissivity of heat structure. Optional.

(type = character*16, default = none, units = none)

(9) ICFCNV

Control function name for convective heat transfer coefficient to pool in reactor cavity CV. Optional.

(type = character*16, default = none, units = none)

Example

```
COR_LHD    5   4  ! NLHT  NLHTA
!         IS  TLH      RADLH       NameCVH   NameHS    Side VF
          1   560.    1.69257      CV200
          2   560.    2.39365      CV200     HS12000   LHS  0.7
          3   560.       R3        CV200     !The user has indicated ring radius 3
          4   560.    4.00000      CV200
          5   560.    4.00000      CV200
```

COR_LHN – Lower Head Nodalization Record

Optional.

This optional record is used to define the mesh nodalization and composition in the lower head and lower vessel cylinder. If this record is absent, the lower head and lower vessel cylinder will each consist of NLH-1 mesh layers of carbon steel. Each will have a thickness of DZLH/(NLH-1) in the lower head and a thickness of DZRV/(NLH-1) in the lower vessel cylinder. If this record is included, then all the mesh layers (from 1 to NLH–1) must be described, beginning from the inner surface of the lower head and proceeding to the outer surface.

MELCOR 1.8.6 requires input for both the lower head and the lower vessel cylinder and permits input for only a single mesh on each record row. There must be a total of NLH-1 record rows included.

If a material defined by this record is not included in the MP package database or if the MP database does not contain all of the required properties for a default material, then the user must provide MP input for the required properties: thermal conductivity, specific heat, density, enthalpy versus temperature, temperature versus enthalpy, melt temperature, and latent heat of fusion for the material.

(1) NML

The lower head mesh layers (number of temperature nodes in the vessel lower head NLH minus one)

(type = integer, default = none, units = none)

The following data are input as a table with length NML:

(1) NL

Index for the lower head mesh layer

(type = integer, default = none, units = none)

(2) MATLH

Material name from the MP package database or a user-defined material that is present in this mesh of the lower head.

(type = character*24, default = none, units = none)

(3) DZHEAD

Distance from inner surface of lower head to outer boundary of this mesh layer. The last value of DZHEAD must be equal to DZLH from record COR_VP.

(type = real, default = none, units = m)

(4) MATCYL

Material name from the MP package database or a user-defined material that is present in this mesh of the lower vessel cylinder.

(type = character*24, default = none, units = none)

(5) DZWALL

Distance from inner surface of lower head to outer boundary of this mesh layer. The last value of DZWALL must be equal to DZRV from record COR_VP.

(type = real, default = none, units = m)

Example

```
COR_LHN 4 !NL MATLH           DZHEAD    MATCYL          DZWALL
           1  STAINLESS-STEEL 0.0254    STAINLESS-STEEL 0.0254
           2  CARBON-STEEL    0.0754    CARBON-STEEL    0.0754
           3  CARBON-STEEL    0.1754    CARBON-STEEL    0.1754
           4  CARBON-STEEL    0.2254    CARBON-STEEL    0.2254
```

COR_LHF – Lower Head Failure Modeling Parameters

Optional.

This record specifies parameters needed to model failure of the vessel lower head and its penetrations. This record is not required, but if included, the following six fields must be present. The default heat transfer coefficients are order-of-magnitude parameters that should be varied in sensitivity studies to determine their impact on lower head heat transfer and failure. The heat transfer coefficients may also be calculated as the values of real control functions or, for the molten pool, from an internal model. The default failure temperature is an approximate value for the transition to plastic behavior for steel. For more information on how these parameters are used in the lower head failure model, see the COR Package Reference Manual, Section 4.

(1) HDBPN

Heat transfer coefficient from debris to penetration structures.

(type = real, default = 1000.0, units = W/m^2-K)

or

(1) ICFHPN

User name of control function defining the heat transfer coefficient (in W/m^2-K) from debris to penetration structures.

(type = character*16, default = none, units = none)

(2) HDBLH

Heat transfer coefficient from debris to lower head.

(type = real, default = 1000.0, units = W/m^2-K)

or

(2) ICFHLH

User name of control function defining the heat transfer coefficient (in W/m^2-K) from debris to the lower head.

(type = character*16, default = none, units = none)

or

(2) MDHLH

Option for heat transfer coefficient from debris to lower head:

(a) MODEL

The coefficient is to be calculated from the internal model.

(type = character*5, default = none, units = none)

(3) HMPOLH

Heat transfer coefficient from the oxidic molten pool to the lower head.

(type = real, default = 1000.0, units = W/m^2-K)

or

(3) ICFHMPO

User name of control function defining the heat transfer coefficient (in W/m^2-K) from the oxidic molten pool to the lower head.

(type = character*16, default = none, units = none)

or

(3) MDHMPO

Option for heat transfer coefficient from the oxidic molten pool to the lower head:

(a) MODEL

The coefficient is to be calculated from the internal model

(type = character*5, default = none, units = none)

(4) HMPMLH

Heat transfer coefficient from the metallic molten pool to the lower head.

(type = real, default = 1000.0, units = W/m^2-K)

or

(4) ICFHMPM

User name of control function defining the heat transfer coefficient (in W/m^2-K) from the metallic molten pool to the lower head.

(type = character*16, default = none, units = none)

or

(4) IHMPMLH

Option for heat transfer coefficient from the metallic molten pool to the lower head:

(a) MODEL
the coefficient is to be calculated from the internal model.

(type = character*5, default = none, units = none)

(5) TPFAIL

Failure temperature of the penetrations or the lower head.

(type = real, default = 1273.15, units = K)

(6) CDISPN

Discharge coefficient for ejection of debris through failed penetration opening.

(type = real, default = 1.0, units = none)

Example

```
!       HDBPH/CF HDBLH/CF HMPOLH/CF HMPMLH/CF TPFAIL     CDISPN
COR_LHF 1.00E-02 1.00E+02 MODEL     MODEL     1525.0     1.0
```

COR_PEN – Lower Head Penetration Record

Required if NPNTOT > 0.

This record specifies information for all representative lower head penetrations. For a description on how these parameters are used in the lower head model, see the COR Package Reference Manual, sections 5.1 and 5.2. This record must be input for each representative penetration from 1 to NPNTOT. If the field IPNREF contains a valid

reference penetration, only the field, specifying the radial ring number, will be processed. If the field IPNREF does not contain a valid reference penetration, all the remaining fields must be present. Total values for all structures represented by this penetration should be supplied.

(1) NPNTOT

Total number of representative lower head penetrations modeled.

(type = integer, default = none, units = none)

The following data are input as a table with length NPNTOT:

(1) NP

Penetration index used for sequencing.

(type = integer, default = none, units = none)

(2) IPNREF

Reference penetration number. IPNREF must be less than NP. Enter 0 if no reference penetration is needed.

(type = integer, default = none, units = none)

(3) IRP

Segment where this representative penetration is located. The value of IRP on this field must be greater than or equal to the value of IRP entered on the previous penetration data strings.

(type = integer, default = none, units = none)

(4) XMPN

Total mass of structures represented by this penetration.

(type = real, default = none, units = kg)

(5) TPN

Initial temperature of representative penetration.

(type = real, default = none, units = K)

(6) ASPN

Total surface area of structures represented by this penetration, based on the height DZ of the bottom axial level. The area of contact with debris is calculated from ASPN based on the debris height.

(type = real, default = none, units = m^2)

(7) AXPN

Total effective conduction area of structures represented by this penetration at junction with the lower head, used to control heat transfer between head and penetration.

(type = real, default = none, units = m^2)

(8) DFLPN

Initial diameter of failure opening for ejection of molten core materials (may represent single or multiple tube failures).

(type = real, default = none, units = m)

(9) ICFLHF

If ICFLHF is input, a value of .TRUE. calculated by a logical control function with user name ICFLHF will trigger failure of the lower head penetrations in this radial ring.

Input NO if a control function is not used.

Optional field.

(type = character*16, default = NO, units = none)

Example

```
COR_PEN   2
!IPNREF IRP  XMPN   TPN    ASPN   AXPN      DFLPN      ICFLHF
 1 0    1    120.0  560.0  15.0   1.5E-02   3.4E-02    PenFailure
 2 1    2
```

2.6 MELGEN Arbitrary Conduction or Radiation Heat Transfer Path

The heat transfer paths modeled within the COR package are appropriate for conventional commercial LWRs. However, paths that are not important in such a reactor may be significant in other applications of MELCOR. The capability has been added to allow the user to define arbitrary additional heat transfer paths between components on the COR_HTR input record. The two components need not be in the same core cell. The modeling may be based on radiation or conduction; in either case, the user is responsible for defining a single input parameter that captures the geometry of the heat transfer path.

COR_HTR – Added COR Heat Transfer Path

Optional.

This record defines additional heat transfer paths within the COR package, between specified components in specified core cells. Such a path may be used to calculate radiation or conduction.

This feature has been extended to allow specification of a heat transfer path from a COR component to a heat structure. The heat transfer path must be defined 'From' a valid COR component and the heat structure must not have a user specified boundary condition (i.e., IBCL = 0,20,30,80, or 90). Furthermore, if a radiation path is defined, the emissivity must be defined by the user on the appropriate HS Boundary Surface Radiation Data record (HS_LBR or HS_RBR).

(1) NHTR

Number of arbitrary conduction or radiation heat transfer paths to be defined.

(type = integer, default = none, units = none)

The following data are input as a table with length NHTR:

(1) N

Path index

(type = integer, default = none, units = none)

(2) IAFROM

Axial level number containing the first participating component.

(type = integers, default = none, units = none)

(3) IRFROM

Radial ring number containing the first participating component.

(type = integers, default = none, units = none)

(4) ICMP1

Two-character identifier for the first component. Depending on the reactor type, the following identifiers are permitted values:

(a) FU
(b) CL
(c) CN
(d) CB
(e) PD
(f) SS
(g) NS
(h) PB
(i) RK
(j) CH
(k) FM
(l) PT
(m) CT

(type = character*2, default = none, units = none)

(5) IATO or HS number

Axial level number containing the second participating component or the user defined heat structure number.

(type = integers, default = none, units = none)

(6) IRTO or HS side

Radial ring number containing the second participating component. If this path transfers to a heat structure, then this field must be either 'LEFT' (or -1) for the left side of a heat structure or 'RIGHT' (or 1) for the right side.

(type = integers or character, default = none, units = none)

(7) ICMP2

Two-character identifier for the second component, specified as for ICMP1 except that HS is permitted for this field.

(type = integer / character*2, default = none, units = none)

(8) HTRMDL

Type of heat transfer path:

(a) 1 or RADIATE-CONST
(b) 2 or CONDUCT-CONST
(c) 3 or RADIATE-CF
(d) 4 or CONDUCT-CF

(type = integer / character*14, default = none, units = none)

optionally followed by

in case HTRMDL = RADIATE-CONST

(9) VFA

Product of view factor and area for this radiative path.

(type = real, default = none, units = none)

in case HTRMDL = CONDUCT-CONST

(9) CND

Total conductance for this conductive path. It represents an effective value of kA/ℓ and, when multiplied by the temperature difference, should give the total heat flow in W.

(type = real, default = none, units = W/K)

in case HTRMDL = RADIATE-CF

(9) ICFVFA

User name of control function defining the product of view factor and area (in m^2) for this radiative path.

(type = character*16, default = none, units = none)

in case HTRMDL = CONDUCT-CF

(9) ICFCND

User name of control function defining the total conductance (in W/K) for this conductive path.

(type = character*16, default = none, units = none)

Example

```
COR_HTR 1  !From: IA IR IC To: IA IR IC FLAG          COEFF
               1  2  4 SS      3  3 SH CONDUCT-CONST 0.0818
```

2.7 MELGEN COR Material Input

In MELCOR 1.8.5, the FU components could contain only UO_2 and (optionally) the heater rod material used in simulation of electrically heated experiments. MELCOR 1.8.6 (and later) allows FU to contain an additional material from the list of materials already known to the COR package. If this material is Zircaloy or steel, FU is also allowed to contain the corresponding oxide, ZrO_2 or steel oxide, respectively.

COR_FUM – Extra Fuel Material Record

Optional.

This record allows specification of an additional material included in the composition of the FU component. If absent, FU can contain only UO_2 and, optionally, the heating element material defined by MATHT on input record COR_RT COR00002.

(1) XFUMAT

Identifier for material in the COR package. Permitted values are:

(a) ZR
(b) ZRO2
(c) SS
(d) SSOX
(e) CRP
(f) INC
(g) GRAPH

(type = character*4, default = none, units = none

If XFUMAT is ZR or SS, the corresponding oxide, ZRO2 or SSOX, will also be allowed as a material in FU.

Example

```
COR_FUM  CRP
```

MELCOR 1.8.5 provided a limited flexibility in the definition of materials by allowing the user to specify the name of the material whose properties were to be used for ZR, etc. However, the change was global in the sense that the same properties would be used in all components (e.g., ZR in CL, CN, and CB). Further, the manual warned that, "Substitution of materials will have no effect on chemical properties assumed for the materials in the oxidation or eutectics models or on any properties used for them in the FDI or CAV packages."

MELCOR 1.8.6 (and later) allows greater flexibility: the user has the option of specifying the primary material in CL, CN/CB, and/or SH from the list of materials known to MELGEN, which includes UO2, ZR, ZRO2, SS, SSOX, and CRP.

MELCOR 2.0 adds GRAPH to the list of known materials.

If that material is ZR, SS, or GRAPH, its oxidation will be correctly modeled; otherwise, no oxidation will be calculated. The eutectics models will treat them appropriately, and the correct properties will be used for them in the FDI and CAV packages.

One possible use of this flexibility is to model an experimental facility such as QUENCH using the PWR option, with the further specification that the SH component is Zircaloy.

COR_CLM – Cladding Material Record

Optional.

This record allows specification of the primary material in the CL component. If this record is absent, that material will be Zircaloy.

(1) CLMAT

Identifier for material in the COR package. Permitted values are:

(a) ZR
(b) ZRO2
(c) SS
(d) SSOX
(e) CRP
(f) INC
(g) GRAPH.

(type = character*4, default = ZR, units = none)

If CLMAT is ZR, SS, or GRAPH, oxidation of this material will be modeled, creating ZRO_2 or SSOX, respectively, for the first two.

Example

```
COR_CLM  SSOX
```

COR_CNM – Canister Material Record

Optional.

This record allows specification of the primary material in the CN and CB components. If this record is absent, that material will be Zircaloy.

(1) CNMAT

Identifier for material in the COR package. Permitted values are:

(a) ZR
(b) ZRO2
(c) SS
(d) SSOX
(e) CRP
(f) INC.

(type = character*4, default = ZR, units = none)

If CNMAT is ZR or SS, oxidation of this material will be modeled, creating ZRO_2 or SSOX, respectively.

Example

```
COR_CNM  SS
```

COR_SHM – Core Shroud Material Record

Optional.

This record allows specification of the primary material in the SH component. If this record is absent, that material will be steel.

(1) SHMAT

Identifier for material in the COR package. Permitted values are:

(a) ZR
(b) ZRO2
(c) SS
(d) SSOX
(e) CRP
(f) INC.

(type = character*4, default = SS, units = none)

If SHMAT is ZR or SS, oxidation of this material will be modeled, creating ZRO_2 or SSOX, respectively.

Example

```
COR_SHM  SS
```

COR_RFM – Reflector Material Record

Optional

This is a new record recognized for the PBR and PMR COR types, which can be used to specify the material for the Reflector component.

(1) RFMAT

Identifier for material in the COR package. Permitted values are:

(a) ZR
(b) ZRO2
(c) SS
(d) SSOX
(e) CRP
(f) INC
(g) GRAPH.

(type = character*4, default = GRAPH, units = none)

Example

```
COR_RFM   'GRAPH'
```

COR_MAT – Core Material Specification Record

Optional.

The materials to be used for control rod poison and electric heater rod material may be specified on the COR_RT input record. These optional records allow the user to redefine the properties to be used for other materials in the COR package. Only the thermal equation of state (including melting point and heat of fusion), density, thermal conductivity, and—for steel—composition will be used. The intention is to allow redefinition of properties without affecting the properties of default materials that may be used in other packages, particularly in HS. Substitution of materials will have no effect on chemical properties assumed for the materials in the oxidation or eutectics models or on any properties used for them in the FDI or CAV packages.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NM

Table record index for material input.

(type = integer, default = none, units = none)

(2) CORMAT

Identifier for material in the COR package. Permitted values are UO2, ZR, ZRO2, SS, and SSOX.

(type = character*4, default = none, units = none

(3) MATNAM

Material name from the MP package database or a user-defined material whose properties are to be used for the designated COR material.

(type = character*24, default = none, units = none

In the absence of input on the COR_MAT record, the properties used will be those of default materials in the MP package: URANIUM-DIOXIDE for UO_2, ZIRCALOY for ZR, ZIRCONIUM-DIOXIDE for ZRO_2, STAINLESS-STEEL for SS, and STAINLESS-STEEL-OXIDE for SSOX.

Example

```
COR_MAT   1 ! CORMAT      MATNAM
          1   UO2         ZIRCONIUM-OXIDE
```

2.8 MELGEN CVH Fluid Flow Interface Input

COR_TIN – Hydrodynamic Control Volume Name Containing Core Cells.

Permitted only if IDTDZ = 1 on record COR_MS.
If IDTDZ = 1: Required for top and bottom volumes containing core cells.
Optional for all other control volumes.

This record allows the user to specify an inlet temperature boundary condition for the dT/dz model that differs from the default modeling. For details on this model, see the COR Package Reference Manual, Section 2.5. If IDTDZ = 1 on record COR_MS, a COR_TIN record is required for every hydrodynamic control volume that (1) contains the lowest or highest axial core cells, or (2) is directly above or below a control volume in which dT/dz calculations are disabled by some other COR_TIN record. The record is optional for all other control volumes containing core cells.

(1) NUMSTR

Number of table records (maximum = NVTOT).

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) IV

Table row number for sequencing.

(type = integer, default = none, units = none)

(2) IVOL

Core control volume name.

(type = character*16, default = none, units = none)

(3) ITUP

Inlet temperature specification for volume IVOL for upflow.

(a) 0
Use exit temperature for next lower control volume. (This is the default for all volumes not appearing on the COR_TIN record.)

(b) 1000
Disable dT/dz model for this volume.

(c) 2000
Use the model employed for IDTDZ = 0. (Must also specify ITDN = 2000.)

(d) 1 or CV
Use temperature of volume ITUPNAM.

(e) -1 or CF
Use value of control function ITUPNAM.

(type = integer / character*2, default = 0, units = none)

(4) ITUPNAM

Control volume name (if ITUP = CV).

Control function name (if ITUP = CF).

(type = character*16, default = none, units = none)

(5) ITDN

Inlet temperature specification for volume IVOL for downflow.

(a) 0
Use exit temperature for next higher control volume. (This is the default for all volumes not appearing on COR_TIN record.)

(b) 1000
Disable dT/dz model for this volume.

(c) 2000
Use the model employed for IDTDZ = 0.

(d) 1 or CV
Use temperature of volume ITDNNAM.

(e) -1 or CF
Use value of control function ITDNNAM.

(type = integer / character*2, default = 0, units = none)

(6) ITDNNAM

Control volume name (if ITDN = CV).

Control function name (if ITDN = CF).

(type = character*16, default = none, units = none)

Example

```
COR_TIN  3 !IV IVOL   ITUP  CV/CF   ITDN  CV/CF
            1  CV120  1000         2000
            2  CV110  CF    CF45   2000
            3  CV111  2000         CV    CV120
```

COR_RAF – Ring Flow Area Control Functions

Optional.

This optional record identifies the control functions (ICFVOL) that may be used in the COR oxidation model to define the flow areas associated with each COR ring within CVH volumes interfaced with the COR package. The ring flow areas are used solely to allocate steam and oxygen from the CVH volumes to the various rings modeled by the COR package.

As specified on input record COR_RP, the core is partitioned into NRAD rings and interfaced with NVTOT CVH volumes (NVTOT is calculated from the set of CVH volumes defined on the COR_RBV record). Each data string on the COR_RAF record defines a control function for control volume IVOL in radial ring IRING and contains entries for three essential fields:

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) IVOL

Core control volume name for which the control function on this data string apply.

(type = character*16, default = none, units = none)

(3) IRING

Radial ring number.

(type = integer, default = none, units = none)

(4) ICFVOL

The name of the REAL control function from which the unblocked flow area of ring IRING is to be obtained.

(type = character*16, default = none, units = none)

Control volume ICVOL may be interfaced with one or more core rings. (In general, a core CVH control volume need not be interfaced with all rings of the core.) In this case, the calculation of unblocked flow area in the ring may be defined by control function ICFVOL. Otherwise, the default calculation is used.

Example

```
COR_RAF 4 !IVOL              IRING  ICFVOL
        1  CV1                2      CF1
        2  CV1                3      Cfunc4
        3  Cvol3              1      'Control function 5'
        4  CVH4               3      'CF 3'
```

COR_FLDIR – Set flow direction by CV for dTdz model

Optional.

This optional record specifies the flow direction for the dTdz model by CV. It provides more geometry information to the dTdz model than the previous default, which was to use the net FROM mass flow for flowpaths in the top core CV in a ring to specifiy the flow direction. The flow direction is set in a CV by specifying the flowpath FL to use, and whether a positive flow in that FL is to be considered UP or DOWN.

(1) NN

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NN:

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) CVName

Core control volume name for which flow direction is to be set.

(type = character*16, default = none, units = none)

(3) FLName

Flowpath name whose mass flow is to be used to specify flow direction in CVName.

(type = character*16, default = none, units = none)

(4) UPDWN

Flag indicating whther a positive mass flow in FLName indicates UP or DOWN flow.

(type = character*16, default = none, units = none)

Core cells associated with CVName will use this flow direction to set the assumed direction of flow in the cells if the dTdz model is on.

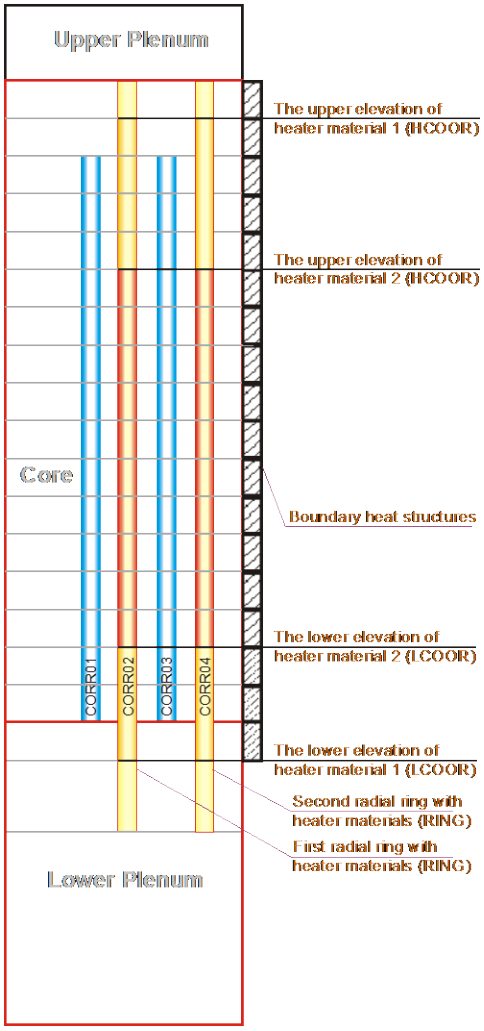
Example

```
COR_FLDIR 4 !CVName     FLName       UPDWN
        1  CV110       FL110        UP
        2  CV111       FL111        UP
        3  CV112       FL112        UP
        4  CV210       FL181-RISER DOWN
```

2.9 MELGEN Electric Heating User Input

The following input records are required to specify an electric heating. A complete set of input records must be supplied. The user input records described below must be set inside the block for the COR package.

This section contains a detailed description of the MELGEN user input for the heater experiments calculations.



COR_ELPOW – Power and Voltage Data

Optional.

This record specifies the power and voltage data in the radial rings with heater materials and contact resistance for the radial rings. Voltage data are

(1) NRING

Number of radial rings with heater materials. Currently the only allowed value is 2.

(type = integer, default = none, units = none)

The following data are input as a table with length NRING:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) RING

The radial ring number with the heater materials.

(type = integer, default = none, units = none)

(3) IPOWF

The key which specifies the source of ring RING power data.

(a) EDF or 2

External data file NAMEFP (input below) is used for ring power data.

(b) TF or 1

Tabular function value NAMEFP (input below) is used for ring power data.

(type = character*3 or integer, default = none, units = none)

(4) NAMEFP

Name of external data file or tabular function (depending on the value IPOWF), which defines the ring power data.

(type = character*16, default = none, units = none)

(4a) NCHANP

The number of channel in external data file NAMEFP, which defines the ring power data. This field is required if and only if IPOWF = EDF or IPOWF = 2.

(type = integer, default = none, units = none)

(5) CONRES

Control function name for total contact resistance definition (is equal to contact resistance per rod divided by number of rods) in radial ring RING.

(type = character*16, default = none, units = none)

The last fields are optional and may be omitted:

(6) IVOLF

The key that specifies the source of ring RING voltage data.

(a) EDF or 2

External data file NAMEFV (input below) is used for ring voltage data.

(b) TF or 1

Tabular function value NAMEFV (input below) is used for ring voltage data.

(type = character*3 or integer, default = none, units = none)

(7) NAMEFV

Name of the external data file or tabular function (depending on the value IPOWF), which defines the ring voltage data.

(type = character*16, default = none, units = none)

(7a) NCHANV

The number of channel in external data file NAMEFV, which defines the ring voltage data. This field is required if and only if IVOLF = EDF or IVOLF = 2.

(type = integer, default = none, units = none)

Example

```
COR_ELPOW 2 ! the number of rings to which power is applied
!Ring number  Source of Power data Contact resistance Source of voltage data
 1 2        EDF EDF20  1     CF20            EDF  EDF21 1
 2 3        EDF EDF20  2     CF20            EDF  EDF21 2
```

COR_ELMAT – Heater Materials Data

Optional, permitted only if COR_ELPOW record is present

This record specifies the electrical properties of heater and electrode materials. By default, the properties of materials HTR, ELM1, and ELM2 are given by DEFW, DEFMO, and DEFCU, respectively, and there is no need for input if this is the desired result.

(1) NMAT

Number of entries in table.

(type = integer, default = none, units = none)

The following data are input as a table with length NMAT:

(1) IMAT

Index in table

(type = integer, default = none, units = none)

(2) CORMAT

CORE material in COR_KFU input

(a) HTR

Material specified as MATHT on record COR_RT

(b) ELM1

Material specified as MATEL1 on record COR_RT

(c) ELM2

Material specified as MAEL2 on record COR_RT

(3) IFLRES

The key that specifies the source of resistivity data for the heater material CORMAT.

(a) DEFCU

Resistivity is defined as default resistivity for CU: RESIST = 1.0E-6*(-7.89E-3 + T*(9.90E-5 + T*(-5.49E-8 + T*3.16E-11))), where T is temperature.

(b) DEFMO

Resistivity is defined as default resistivity for Mo: RESIST = 1.0E-6*(2.29E-2 + T*(5.36E-5 + T*(1.38E-7 - T*2.22E-11))), where T is temperature.

(c) DEFW

Resisitivity is defined as default resistivity for W: RESIST = 1.0E-6*(-2.61E-2 + T*(2.63E-4 + T*2.20E-8)), where T is temperature.

(d) TF

Resistivity is defined as function of temperature from tabular function NAMEFR value (input below).

(type = character*5, default = none, units = none)

If IFLRES is TF, this must be follwed by

(3a) NAMEFR

Tabular function name that defines the resistivity for the heater material CORMAT.

(type = character*16, default = none, units = none)

Example

```
! Change the resistivity of MATHT from 'DEFW' to TF 'Resist_TF'
COR_ELMAT 1  ! Number of table entries
!         INDX  CorName  Resistivity  TF_Name
!         1     HTR      TF           Resist_TF
```

2.10 MELGEN Edit Option Control Input

In MELCOR 1.8.6 (and later), the options that control the presence or absence of sections of the print and plot edit are set in MELGEN and are preserved as part of the restart file. The COR_EDV record, which modifies the default settings, may now be read by MELGEN as well as by MELCOR.

The treatment of IPMV (field 5) has been modified. It only affects the presence of detailed component masses in the plot file; core component volume fractions will always be present.

COR_EDV – Edit Flags Record

Optional.

This record specifies edit flags that allow the user to control which variables from the COR package are edited to the print output file. This record is not required and if it is present, any number of flags up to six may be present. The flag for a particular edit block should be set to 1 if printing of that block is desired and to 0 if not.

(1) ITEMP

Temperature edit flag.

(type = integer, default = 1, units = none)

(2) IMASS

Mass edit flag.

(type = integer, default = 1, units = none)

(3) IVOL

Volume edit flag.

(type = integer, default = 0, units = none)

(4) IASUR

Surface area edit flag.

(type = integer, default = 0, units = none)

(5) IPMV

Component masses plot flag.

(type = integer, default = 0, units = none)

(6) IPOW

Decay heat/fission power edit flag.

(type = integer, default = 0, units = none)

Example

```
COR_EDV   1 1 1 1 1 1
```

2.11 MELCOR Input

Records COR_RF, COR_TP, COR_CHT, COR_CMT, COR_CCT, COR_LP, and record COR_LHF described in Section 2 may also be included in MELCOR input for any restart. These records input basic modeling parameters for the COR package, and it is desirable to have the capacity to vary these parameters for testing purposes and perhaps some sensitivity analyses. However, the user should be aware that changing these parameters in the middle of a calculation could lead to nonphysical results.

The COR_EDV record may be included in MELCOR input to alter the options for presence or absence of sections of the print and plot edit set in MELGEN (or in a previous execution of MELCOR).

Additional records that may be input for the COR package for a MELCOR restart run are concerned with timestep control, package edit control, messaging, and physics module disabling. These records are described below:

COR_DTC – COR Package Timestep Control Record

Optional.

This record allows the user to specify certain quantities needed to control the COR package internal timestep. This record is not required, but if included, two floating point fields and one integer field must be present.

(1) DTMPCR

Maximum allowed temperature increase of a component during a COR package timestep. This maximum is applied only to components with total mass greater than the minimum specified in sensitivity coefficient array 1502.

(type = real, default = 30.0, units = K)

(2) DTCMIN

Minimum COR package timestep allowed.

(type = real, default = 0.001, units = s)

(3) NSUBMX

Maximum number of COR package subcycles during a single system cycle.

(type = integer, default = 64, units = none)

Example

```
COR_DTC 40. 0.05  80
```

COR_TST – Physics On/Off Switches

Optional.

This record specifies switches that allow the user to disable portions of the COR package physics for purposes of debugging or testing of new physics modules. This record is not required, and any number of integer fields may be present, up to 15. Set these switches to 1 to disable a particular model and to 0 to re-enable.

(1) IRAD

A value of 1 disables the radiation model, including radiation to steam, (regardless of the value of switch number 8).

(type = integer, default = 0, units = none)

(2) ICND

Disable switch for conduction model.

(type = integer, default = 0, units = none)

(3) ICNV

Disable switch for convection model.

(type = integer, default = 0, units = none)

(4) IOXD

Disable switch for all oxidation models, including the B_4C reaction (regardless of the value of switch number 7).

Note: There are two oxidation options: (1) the default option for hierarchical treatment in which steam oxidation does not begin until all oxygen has been consumed, and (2) an option for treating oxidation by oxygen and steam simultaneously. Furthermore, oxidation of the conglomerate debris may be deactivated without deactivating the oxidation of intact components. The following values of IOXD may be specified:

(a) 0

(default) hierarchical option for both intact components and conglomerate debris:

(b) 1

No oxidation of either intact components or conglomerate debris

(c) 2
Hierarchical option for intact components with no conglomerate debris oxidation.

(d) 3
Simultaneous option for both intact components and conglomerate debris

(e) 4
Simultaneous option for intact components with no conglomerate debris oxidation.

(type = integer, default = 0, units = none)

(5) IDRP

Disable switch for candling model.

Note: A value of IDRP=1 completely disables the candling model while a value of IDRP=2 disables only the melt holdup model.

(type = integer, default = 0, units = none)

(6) ITDZ

Disable switch for core dT/dz model.

(type = integer, default = 0, units = none)

(7) IB4C

Disable switch for B_4C reaction model, if not already disabled by switch 4.

(type = integer, default = 0, units = none)

(8) IRDS

Disable switch for radiation to steam model, if not already disabled by switch 1.

(type = integer, default = 0, units = none)

(9) IDEJ

Disable switch for solid debris ejection model.

(type = integer, default = 0, units = none)

(10) ISPR

Note: A value of ISPR=1 disables only the molten material radial relocation; a value of ISPR=2 disables only the particulate debris radial relocation model; and a value of ISPR=3 disables both radial relocation models.

Disable switch for radial relocation model.

(type = integer, default = 0, units = none)

(11) IOXS

Disable switch for oxidation of unquenched submerged surfaces, where not already disabled by switch 4.

(type = integer, default = 0, units = none)

(12) IMPMOD

A value of 1 disables inclusion of molten pool models..

(type = integer, default = 0, units = none)

(13) IMPHT

A value of 1 disables treatment of molten pool heat transfer.

(type = integer, default = 0, units = none)

(14) IMPSTR

A value of 1 disables treatment of molten pool stratification.

(type = integer, default = 0, units = none)

(15) IMPPRT

A value of 1 disables partition of RNs between stratified molten pools.

(type = integer, default = 0, units = none)

(16) IMPLZDM

A value of 1 enables Lipinski zero-dimensional correlation to calculate the drayout heat flux.

(type = integer, default = 0, units = none)

Example

```
COR_TST   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

COR_MSG –Enable Collapse Messaging

Optional

This record enables component collapse messages to be print to the standard output indicating collapse of components at each COR cell. This record is not required, and collapse messages will be disabled when this record is absent.

2.12 Required Input to Other MELCOR Packages

Many of the models in the COR package require that it interface with other MELCOR packages. Thus, in order for the COR package to function properly, input may be required in other packages to complete these interfaces. This section outlines these input requirements for the CVH, FL, NCG, Control Function (CF), Decay Heat (DCH), RN, and HS packages. The user is referred to the appropriate users' guides and reference manuals for further details regarding these packages.

2.12.1 CVH, FL, and NCG Input

The COR package models heat transfer between core structures and the liquids and/or gases that flow through the core. These flows are modeled by the CVH and FL packages. The user must specify a control volume for the coolant channel (and for the bypass for BWRs) in each core cell in order for that cell to communicate heat transfer information. This is done through COR input record COR_RBV, as described in Section 2.4. Each control volume referenced on this record must be defined in the CVH input. The control volumes must be interconnected by flow paths that are defined through input to FL. COR only checks that each control volume referenced by COR input exists. The user is responsible for providing CVH and FL input that is reasonable, logical, and above all, consistent with the physical realities of the core geometry to be modeled.

Some degree of consistency is required between the distribution of fluid volumes implied by input to COR and CVH. The specific requirement is that the initial total fluid volume in COR lying within any V/A segment of a CVH control volume cannot exceed the fluid volume in that V/A segment, so that filling the entire fluid volume in COR with debris cannot overfill CVH. This requirement is new in MELCOR 1.8.5. See Section 1.5.3 for suggestions on revising old input decks to comply.

Another COR input, which is found on record COR_TIN, is indirectly related to CVH and FL in that it is used to determine the inlet temperature to the core whenever the dT/dz model is used to estimate local hydrodynamic temperatures within the core. The user must ensure that any control volume or control function referenced by these records exists in the CVH or CF input.

If the user desires to perform detailed natural circulation calculations using the CVH and FL packages, one CVH control volume should be specified for a relatively small number of core cells (perhaps only one cell per volume). The core flow blockage and momentum flux models should be enabled for all flow paths involving core cells, as described in the FL Package Users' Guide (input records FL_BLK and FL_MFX). Note that new blockage options, including a channel box model, are available in MELCOR 1.8.5. The dT/dz model should <u>not</u> be disabled in these flowpaths. Radial flow paths may be used in PWRs, and channel-to-bypass flow paths in BWRs. There will probably be some penalty in increased CPU time resulting from the need to invert a substantially larger flow matrix. However, use of the blockage model has been found to largely eliminate the further penalty that would otherwise be associated with reduced material courant timestep limits when control volumes are almost completely filled with core debris.

H_2 and CO must be specified as active materials in the NCG package whenever the COR package is active due to the generation of these gases in the COR oxidation models. In addition, for calculations with B_4C used as the control poison material, the gases O_2, CO_2, and CH_4 must also be specified as active materials via NCG input, since they are produced by the B_4C reaction model.

2.12.2 DCH and RN Input

The COR package distributes the decay heat generated by the fission products and, if applicable, the fission energy to the core components. The properties of the time-

dependent decay heat distribution are specified through input to both the DCH and RN packages. The fission power is specified in the COR package through the use of a control function.

The total core decay heat is set up through input to the DCH package. In general, the user specifies the type of reactor to be modeled, the problem time at which the reactor shut down, the power level at which the reactor was operating, and the type of decay heat correlation to be used.

The decay heat is distributed among the core cells in different ways depending on whether or not the RN package is active. If the RN package is not active, the energy is distributed along with any fission power according to the axial and radial power density profiles specified on records COR_ZP and COR_RP, respectively.

When the RN package is active, the decay heat is distributed throughout MELCOR in proportion to the radionuclide distribution as determined by the RN package. The user must specify the initial distribution of radionuclides to RN through the use of the RN package input records described in Section 3.1.2 of the RN Package Users' Guide. <u>If these records are not used, no decay heat whatsoever may be passed to the core.</u>

In Anticipated Transient Without SCRAM (ATWS) calculations or other problems involving fission heating, whole-core fission power may be calculated using a control function specified on record COR_TP, which must be set up through input to the CF package. The power is distributed (and added to decay heat) in the COR package according to the axial and radial power density profiles specified on records COR_ZP and COR_RP, respectively.

Furthermore, if the advanced B_4C reaction model is invoked, seven additional RN package classes must be defined to treat the additional reaction products generated by the model (see example input in Section 0).

2.12.3 HS Input

In order to calculate the energy radiated from the COR cells, the COR package requires boundary heat structures to be specified for levels above HLST (the top of the portion of the vessel cylinder). These heat structures must be set up through appropriate input to HS, and their existence is checked by the COR package in MELGEN. For the dT/dz model to function properly, <u>there must be exactly one heat structure per axial level above HLST</u>; otherwise, the effect of these heat structures on the control volume temperature distribution will not be treated correctly. Also, no other heat structures may communicate with the core and lower plenum control volumes other than those specified as IHSA on the COR_ZP input record. Furthermore, for the dT/dz model to function correctly and for the HS package to model the heat transfer appropriately, it is important that the heat structures representing the radial core boundary (e.g., core shroud) communicate with the temperatures calculated by the dT/dz model. The outer ring core cells must, therefore, be specified as the fluid temperature boundary on input records HS_LBF and HS_RBF (see the HS Package Users' Guide) unless the IHSDT option switch provided on input record COR_MS has been set to 1 or NOTREQ.

Melting of these boundary heat structures may be modeled using the HS_DG input records. As any such structure melts, the molten steel will be added to core debris in the adjoining core cell. That cell is not specified directly as part of HS input, but is inferred from the elevation of the heat structure; structures above the top of the core deliver their steel to the outermost ring of the uppermost level of the core.

2.12.4 MP Input

The COR package obtains materials properties from the MP package. In addition to general consistency of the various tables and constant properties defining the thermal equation of state, the COR package assumes that the melting range of any pure material spans exactly 0.01K. This is true for all default materials in the MP package, as noted in the introductory paragraphs of Section 2 of the MP Reference Manual.

However, if the properties of a default material used by core are modified, or if a user-defined material is specified to represent a core material, it is essential that the 0.01K melting range be preserved. A user-defined material may be input either as the electric heating element material, MATHT, on the COR_RT input record, or as a substitute for one of the other core materials, as MATNAM on a COR_MAT input record.

3 Sensitivity Coefficients

This section lists the sensitivity coefficients in the COR package that are accessible to the user, along with a brief description, and gives their default values, units, and equivalence names. Sensitivity coefficients in the COR package are grouped as follows:

1001 – 1299	oxidation and heat transfer parameters
1301 – 1399	fission power parameters
1401 – 1499	numerical control parameters
1501 – 1599	geometric parameters
1600 – 1699	lower-head mechanical model parameters

For all user accessible sensitivity coefficients listed in the COR package, the following record is used to override default sensitivity coefficient values. More details are given in the COR Package Reference Manual on the usage of these coefficients in the models where they are applied and the basis for their default values.

COR_SC – COR Sensitivity Coefficients Input Record

Optional.

This record allows the user to change the default COR sensitivity coefficient values to user specified values. This record is not required.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the COR sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

First sensitivity coefficient index.

(type = integer, default = none, units = none)

(5) NB

Second sensitivity coefficient index. Specific sensitivity coefficient dependent.

(type = integer, default = none, units = none)

Example

```
COR_SC  12  ! N   SC          VALUE    INDEX1 INDEX2
              1   1001        1.2E+4        6      2
              2   1003        6.0E-3        1
              3   1004        1.0E-3        1
              4   1005        1.7E3         4
              5   1007        0.3           3      1
              6   1007        0.5           5      2
              7   1007        0.1           1      2
              8   1007        0.9           9      1
              9   1030        0.2           2
             10   1213        8.0E-3        1
             11   1213        1.2E-3        2
             12   1504        1.0E-5        1
```

3.1 Oxidation, Heat Transfer, and Relocation Parameters

1001 – Zircaloy Oxidation Rate Constant Coefficients

These coefficients are used to calculate the rate constant for oxidation of Zircaloy by parabolic kinetics. The rate constant K (kg^2/m^4-s) as a function of temperature T (K) is calculated by

$$K(T) = C1001(1,I)\exp(-C1001(2,I)/T), T \le C1001(5,I)$$

$$K(T) = C1001(3,I)\exp(-C1001(4,I)/T), T \ge C1001(6,I)$$

where I=1 for oxidation by H_2O and I=2 for oxidation by O_2. An interpolated value is used in the temperature range C1001(5,I) < T < C1001(6,I).

(1,I)	Low temperature range constant coefficient. (default = 29.6 for I=1, 26.7 for I=2; units = $kg^2(Zr)/m^4$-s, equiv = none)
(2,I)	Low temperature range exponential constant. (default = 16820.0 for I=1, 17490.0 for I=2; units = K, equiv = none)
(3,I)	High temperature range constant coefficient. (default = 87.9 for I=1, 26.7 for I=2; units = $kg^2(Zr)/m^4$-s, equiv = none)
(4,I)	High temperature range exponential constant. (default = 16610.0 for I=1, 17490.0 for I=2; units = K, equiv = none)
(5,I)	Upper temperature boundary for low temperature range. (default = 1853.0 for I=1, 9998.0 for I=2; units = K, equiv = none)
(6,I)	Lower temperature boundary for high temperature range (default = 1873.0 for I=1, 9999.0 for I=2; units = K, equiv = none)

1002 – Steel Oxidation Rate Constant Coefficients

These coefficients are used to calculate the rate constant for oxidation of steel by parabolic kinetics. The rate constant K (kg^2/m^4-s) as a function of temperature T (K) is calculated by

$$K(T) = C1002(1,I)\exp[-C1002(2,I)/T]$$

where I=1 for oxidation by H_2O and I=2 for oxidation by O_2. No oxidation of steel by O_2 is calculated.

(1,I)	Constant coefficient.

(default = 2.42E09 for I=1, 0.0 for I=2; units = kg^2(steel)/m^4-s, equiv = none)

(2,I) Exponential constant.
(default = 4.24E04 for I=1, 0.0 for I=2; units = K, equiv = none)

1003 – Gaseous Diffusion Oxidation Coefficients

These coefficients are used to calculate the oxidation rate for Zircaloy and steel based on the gaseous diffusion of steam through hydrogen to the metal surface. The following equation gives the oxidation rate W (kg/s) in terms of the mass transfer coefficient k_c (m/s), the metal surface area A (m^2), the difference in oxidant partial pressure between the bulk gas and the surface ΔP_{ox} (Pa), the film temperature T_f (K), and the sensitivity coefficient C1003(m):

$$W = k_c A \Delta P_{ox} C1003(m) / T_f ,$$

where C1003(m) is defined in terms of the universal gas constant R, the molecular weight of the metal M_m, and the number of moles of steam to oxidize one mole of metal n_m:

$$C1003(m) = M_m / (n_m \cdot R)$$

The default values may be changed to selectively increase or decrease the gaseous diffusion based oxidation rate.

(1) Zircaloy constant.
(default = 0.00548, units = kg(Zr)-K/Pa-m^3, equiv = none)

(2) Steel constant.
(default = 0.00504, units = kg(steel)-K/Pa-m^3, equiv = none)

1004 – Oxidation Cutoff Temperatures

These coefficients are used to prevent Zircaloy and steel oxidation below or above certain temperatures. The lower cutoff temperature prevents oxidation at temperatures that generate only minute quantities of noncondensible gases. The upper cutoff temperature can be used to limit the amount of oxidation for sensitivity analyses.

(1) Minimum oxidation temperature.
(default = 1100.0, units = K, equiv = TOXMIN)

(2) Maximum oxidation temperature.
(default = 9900.0, units = K, equiv = TOXMAX)

1005 – B_4C Reaction Model Parameters

These coefficients are used to control the B_4C reaction model. If reducing environments are expected (significant hydrogen concentrations), then the advanced model should be selected and the user should refer to Section 0 for an example of the additional required RN and DCH input. Selection of the simple or advanced reaction model is now made on record COR_B4C, and C1005(1) is no longer used.

(1) No longer used. Presence of an input record that attempts to define the value will be treated as a fatal error.

(2) Maximum B_4C fraction that may be consumed by the reaction model.
(default = 0.02, units = none, equiv = none)

(3) Intact steel failure fraction. The B_4C reaction cannot begin until the ratio of the intact steel mass to its initial value falls below this fraction.
(default = 0.9, units = none, equiv = none)

(4) Reaction threshold temperature. The B_4C temperature must exceed this value or the reaction cannot proceed.
(default = 1500., units = K, equiv = none)

1006 – B_4C Reaction Rate Parameters

These coefficients are used to control the reaction rate of B_4C. The fractional reaction rate (per second) is given by.

$$\frac{d(M/M_o)}{dt} = C1006(1)\exp\left(\frac{-C1006(2)}{T}\right)$$

where M_o is the initial mass of the B_4C and M and T are the current mass and temperature of the B_4C, respectively.

(1) Leading coefficient.
(default = 1.662E5, units = s^{-1}, equiv = none)

(2) Exponent.
(default = 2.26472E4, units = K, equiv = none)

1007 – Ring Minimum Flow Area Fractions

These coefficients establish the minimum flow area fractions to be used in the oxidation model to allocate steam and oxygen from the core CVH control volumes to the various rings associated with the control volumes. If the fraction of the CVH volume flow associated with a ring in a control volume falls below the minimum given by the

corresponding element of C1007, the ring is assumed to be blocked for oxidation purposes and is assigned that minimum fraction of the oxidizer inventory. The remaining unassigned flow is divided among the unblocked rings (if there are any). If the sum over the NRAD rings in the core exceeds inventory, an error message will be printed and MELGEN/MELCOR execution terminated.

The minimum fractions may be assigned differently for the channel and bypass volumes in a BWR (when those volumes are distinguished in user input).

(IR,1) Minimum flow area fraction for the channel-side portion of ring IR ($1 \leq$ IR $\leq$ NRAD).
[default=0.0, units=none, equiv=ACHNNL(IR)]

(IR,2) Minimum flow area fraction for the bypass-side portion of ring IR ($1 \leq$ IR $\leq$ NRAD).
[default=0.0, units=none, equiv=ABYPAS(IR)]

1010 – Material Dissolution Rate Coefficients

These coefficients are used to limit the rate of dissolution of materials by parabolic kinetics. If the coefficients are defined for material j, then the mass fraction of that material in a mixture at the end of the timestep cannot exceed

$$XF_j^2 = XI_j^2 + K_j \Delta t$$

where

K_j	=	C1010(1,j) exp(-C1010(2,j)/T)
	=	dissolution constant for material j
XF_j	=	final mass fraction of material j in the mixture
XI_j	=	initial mass fraction of material j in the mixture
Δt	=	timestep size
T	=	temperature (K)

j	Solid	C1010(1,j) (unit = none)	C1010(2,j) (units = K)
1	Zr	-1.0(undefined)	0.0 (undefined)
2	ZrO_2	1.47e14	8.01e04
3	UO_2	1.02e15	8.14e04
4	Steel	-1.0	0.0
5	Steel oxide	-1.0	0.0
6	Control poison	-1.0	0.0

j	Solid	C1010(1,j) (unit = none)	C1010(2,j) (units = K)
7	Inconel	-1.0	0.0

1011 – Eutectic Reaction Temperatures

These coefficients are used to define the temperature at which the rate of eutectic reaction between two contacting solids is sufficiently high to cause significant liquefaction on a time scale of interest. For example, although the Zircaloy-Inconel eutectic reaction may begin at temperature as low as 1200 K, the rate of reaction is insignificant until the temperature reaches about 1400 K.

(1) Zircaloy-Inconel eutectic temperature.
(default = 1400.0, units = K, equiv = TZRINC)

(2) Zircaloy-steel eutectic temperature.
(default = 1400.0, units = K, equiv = TZRSS)

(3) Steel-B_4C eutectic temperature.
(default = 1520.0, units = K, equiv = TSSB4C)

1016 – Zircaloy Post-Breakaway Oxidation Rate Constant Coefficients

These coefficients are used to calculate the rate constant for oxidation of Zircaloy by parabolic kinetics. The rate constant K (kg^2/m^4-s) as a function of temperature T (K) is calculated by

$$K(T) = C1016\,(1,I)\exp(-C1016\,(2,I)/T)$$

where I=1 for oxidation by H_2O and I=2 for oxidation by O_2. Currently, although there is provision for using a breakaway model for oxidation by steam, this is thought to be unnecessary and the steam coefficients are zero.

(1,I) Constant coefficient.
(default = 0 for I=1, 2.97e3 for I=2; units = $kg^2(Zr)/m^4$-s, equiv = none)

(2,I) Exponential constant.
(default = 0 for I=1, 19680.0 for I=2; units = K, equiv = none)

1017 – Lifetime Parameters for Breakaway Model

The lifetime parameter for Zircaloy-air oxidation breakaway is given by

$$P_{LOX} = C1017(1) \cdot \log_{10} T + C1017(2)$$

where T is the clad temperature in K. The default values are from a fit by Randy Gauntt. The time to breakaway in seconds is given by

$$\tau = 10^{P_{LOX}}$$

where T is the temperature in K.

(1) Inherently negative multiplicative constant.
(default = -12.528, units = none, equiv = none)

(2) Inherently positive additive constant.
(default = 42.038, units = none, equiv = none)

1018 – Maximum Lifetime for Breakaway Model

The maximum lifetime used in the lifetime rule is normally 1.0. However, it was found that a smoother change from pre-breakaway to post-breakaway gave a better fit to experimental data. The results of the Spent Fuel Pool Experiment (SFP) suggest that a maximum lifetime of 1.2 gives a better fit for the default breakaway parameters.

(1) Maximum lifetime.
(default = 1.2, units = none, equiv = none)

There are also two control function arguments available: COR-ZROX-TLEFT and COR-ZROX-LIFE (see Section 5.6 for a description of these control functions).

1020 – Radial Relocation Model Parameters

These parameters control the operation of the routines that relocate both molten material and solid particulate debris from ring to ring.

(1) Time constant for the relocation of solid material.
(default = 360.0 [note that significant changes in the default value have occurred], units = s, equiv = TSPRS)

(2) Time constant for the relocation of molten material.
(default = 60.0 [note that significant changes in the default value have occurred], units = s, equiv = TSPRM)

(3) Lower head curvature bias for solid material (currently unused).
(default = 0.0, units = none, equiv = FDIPS)

(4) Lower head curvature bias for molten material (currently unused).

(default = 0.0, units = none, equiv = FDIPM)

(5) Fraction of cell CVH volume that is available for radial relocation.

(default = 1.0, units = none, equiv = CVHFAC)

1021 – Channel-Bypass Relocation Time Constant

This parameter controls the rate of relocation of debris (molten and solid) between the channel and bypass of a BWR after failure of the canisters.

(1) Time constant for the relocation of debris.

(default = 1.0, units = s, equiv = TSPCB)

1030 – dT/dz Model Parameters

These coefficients are associated with the dT/dz model.

(1) Option switch; if nonzero, the treatment reverts to that used in MELCOR 1.8.3 (and earlier versions) where the possibility of down flow was not considered.

(default = 0.0, units = none, equiv = none.)

(2) Time constant for averaging flows. The smoothed flows are also used in determining the flow direction if C1030(1) is zero.

(default = 0.1, units = s, equiv = TRLXZ)

(3) Time constant for relaxing the dT/dz temperatures towards the CVH volume temperature when the dT/dz model is disabled.

(default = 1.0, units = s, equiv = TRLXT)

(4) Characteristic time for coupling dT/dz temperatures to average CVH volume temperatures when the dT/dz model is active.

(default = 10.0, units = s, equiv = none)

(5) Maximum relative weight of old flow in the smoothing algorithm involving TRLXZ=C1030(2).

(default = 0.6, units = none, equiv = none)

1101 – Fuel-Cladding Gap Emissivities

These coefficients define the fuel and cladding surface emissivities.

(1) Fuel surface emissivity.
(default = 0.8, units = none, equiv = EMISF)

(2) Cladding inner surface emissivity.
(default = 0.325, units = none, equiv = EMISC)

1102 – Steel Emissivity

These coefficients define the emissivity of steel. The emissivity is first calculated as

$$\varepsilon_{\text{steel}} = C1102(2) + C1102(3)[T - C1102(1)]$$

and is then bounded to lie in the range of

$$C1102(5) \le \varepsilon_{\text{steel}} \le C1102(4)$$

The default values duplicate the correlations used in MELCOR 1.8.5; the default reference temperature is 650°F, and the default temperature coefficient is 0.000193/°F.

(1) Reference temperature.
(default = 616.4833, units = K, equiv = EMIS_SS_Tref)

(2) Reference value.
(default = 0.25617, units = none, equiv = EMIS_SS_RefVal)

(3) Temperature coefficient.
(default = 0.0003474, units = K^{-1}, equiv = EMIS_SS_Tcoef)

(4) Maximum value.
(default = 0.9999, units = none, equiv = EMIS_SS_Max)

(5) Minimum value.
(default = 0.0001, units = none, equiv = EMIS_SS_Min)

1103 – Particulate Emissivity

These coefficients define the emissivity of particulate debris. The emissivity is first calculated as

$$\varepsilon_{steel} = C1103(2) + C1103(3)[T - C1103(1)]$$

and is then bounded to lie in the range of

$$C1103(5) \le \varepsilon_{\text{steel}} \le C1103(4)$$

The default values duplicate the correlations used in MELCOR 1.8.5, in which a constant value of 0.9999 was used.

(1) Reference temperature.
(default = 1000.0, units = K, equiv = EMIS_Px_Tref)

(2) Reference value.
(default = 0.9999, units = none, equiv = EMIS_Px_RefVal)

(3) Temperature coefficient.
(default = 0.0, units = K^{-1}, equiv = EMIS_Px_Tcoef)

(4) Maximum value.
(default = 0.9999, units = none, equiv = EMIS_Px_Max)

(5) Minimum value.
(default = 0.0001, units = none, equiv = EMIS_Px_Min)

1104 – Oxidized Zircaloy Emissivity

These coefficients define the emissivity of oxidized Zircaloy. The emissivity is first calculated as a function of oxide thickness, Δr_{ox} by linear interpolation in the following table:

Δr_{ox}	ε
0.0	C1104(1)
C1104(2)	C1104(3)
C1104(4)	C1104(5)

If the maximum temperature, that has been reached by the component exceeds C1104(6), this value is reduced by

$$f = exp\left[\frac{C1104(6) - T_{max}}{C1104(7)}\right]$$

The value is then bounded to lie in the range of

$$C1104(9) \le \varepsilon_{Zircaloy} \le C1104(8)$$

The default values duplicate the correlations used in MELCOR 1.8.5 for oxide thicknesses of less than 1 mm.

(1) Clean-surface emissivity.
(default = 0.325, units = none, equiv = EMIS_Zr_Clean)

(2) Oxide thickness 1.
(default = 3.8799999E-06, units = m, equiv = EMIS_Zx_Ox1)

(3) Emissivity at oxide thickness 1.
(default = 0.808448, units = none, equiv = EMIS_Zx_Emis1)

(4) Oxide thickness 2.
(default = 0.001, units = m, equiv = EMIS_Zx_Ox2)

(5) Emissivity at oxide thickness 2.
(default = 0.758642, units = none, equiv = EMIS_Zx_Emis2)

(6) Temperature above which emissivity is degraded.
(default = 1500.0, units = K, equiv = EMIS_Zx_Tdegr)

(7) Temperature range over which emissivity is degraded.
(default = 300.0, units = K, equiv = EMIS_Zx_dTdegr)

(8) Maximum value.
(default = 0.9999, units = none, equiv = EMIS_Px_Max)

(9) Minimum value.
(default = 0.325, units = none, equiv = EMIS_Px_Min)

1105 – Graphite Emissivity

These coefficients define the emissivity of graphite, and were added for HTGRs. The emissivity can be specified for block graphite and matrix material. The coefficients are used by the core radial conductivity functions (see COR_TKE): if the core type is PBR, the matrix emissivity is used for the radial core conduction; if PMR, then the block emissivity is used. Note that the block emissivity for the compact-block gap in a PMR is specified on CS1101(2).

(1) Graphite block emissivity.
(default = 0.8, units = none, equiv = EMIS_Gr)

(2) Matrix material emissivity.
(default = 0.8, units = none, equiv = EMIS_Gr_matrix)

1131 – Molten Material Holdup Parameters

These coefficients are used to define conditions for which molten material will be held up by an oxide shell. The conditions for melt breakout are probably rather different for a canister (CN or CB) than for fuel rod cladding (CL). New elements were added to this sensitivity coefficient array in MELCOR 1.8.6 to allow independent specification. The defaults are currently set so that there is holdup. See the COR Package Reference Manual, Section 3.1, and Section 1.3 of this manual for details on the molten material holdup model.

(1) Minimum ZrO_2 thickness required to hold up molten Zr in CL.

(default = 0.00001, units = m, equiv = DZXMN)

(2) Maximum ZrO_2 temperature permitted to hold up molten Zr in CL.

(default = 2400.0, units = K, equiv = TZXMX)

(3) Minimum steel oxide thickness required to hold up molten steel.

(default = 0.001, units = m, equiv = DSXMN)

(4) Maximum steel oxide temperature permitted to hold up molten steel.

(default = 1700.0, units = K, equiv = TSXMX)

(5) Minimum ZrO_2 thickness required to hold up molten Zr in CN, CB, or SH.

(default = 0.00001, units = m, equiv = DZXMN1)

(6) Maximum ZrO_2 temperature permitted to hold up molten Zr in CN, CB, or SH.

(default = 2100.0, units = K, equiv = TZXMX1)

1132 – Core Component Failure Parameters

These coefficients define the temperatures used in extended failure criteria for fuel after the Zr has melted and candled [by virtue of exceeding the temperature specified by C1131(2)].

In older versions of MELCOR, fuel rods were converted to particulate debris when the remaining thickness of unoxidized Zircaloy in the cladding fell below the value set by DRZRMN on input record COR_CCT. MELCOR 1.8.4 and later versions allow oxidized rods to continue to stand until the temperature reaches TRDFAI. In MELCOR 1.8.4, the default value was 2800 K, the approximate temperature of the UO_2/ZrO_2 eutectic. While this may be appropriate for new fuel, evidence from Phebus suggests that 2500 K is more appropriate for irradiated fuel. The variable TRDMAX defines the absolute maximum temperature at which rods can stand, regardless of Zircaloy content or the value of TRDFAI; the default value is the approximate melting temperature of UO_2. Refer to Section 1.3 of this manual for additional information.

(1) Temperature to which oxidized fuel rods can stand in the absence of unoxidized Zr in the cladding.

(default = 2500.0, units = K, equiv = TRDFAI)

(2) Temperature at which fuel rods will fail, regardless of composition of the cladding.

(default = 3100.0, units = K, equiv = TRDMAX)

1141 – Core Melt Breakthrough Candling Parameters

These coefficients are used to control the candling model when molten material has just been released after holdup by an oxide shell or by a flow blockage (crust). For this case, the quasi-steady assumptions regarding melt generation built into the candling model are not valid, so an alternate timestep size (DTBRK below) is used, along with a maximum melt flow rate (GAMBRK), to eliminate dependencies on molten pool mass and timestep size. See the COR package Reference Manual, Section 3.1.3, for more complete details. The default values are currently chosen so that the model will be active only for large molten pools breaching a crust.

(1) Timestep size used in the candling model for molten material releases immediately after breakthrough of an oxide shell or crust.
(default = 1.0, units = s, equiv = DTBRK)

(2) Maximum melt flow rate per unit width after breakthrough.
(default = 1.0, units = kg/m-s, equiv = GAMBRK)

1151 – Conglomerate Debris Surface Area Coefficients

These coefficients are used in the model that calculates the surface area of conglomerate debris and the portion of the intact component surface area that remains unblocked by the conglomerate debris. These areas are currently used only in the component oxidation models. See the COR Package Reference Manual, Section 3.1.6, for details. The components are, in order, FU, CL, CN, CB, OS, PD, SS, NS, and PB.

(I,1) Maximum surface area fraction for debris rivulets on component I.
[default = 0.556, units = none, equiv = FA1MAX(I)]

(I,2) Maximum surface area fraction for voids in debris for component I.
[default = 0.807, units = none, equiv = FA2MAX(I)]

(I,3) Fraction of interstitial volume corresponding to FA1MAX(I).
[default = 0.143, units = none, equiv = FV1MAX(I)]

(I,4) Fraction of interstitial volume corresponding to FA2MAX(I).
[default = 0.396, units = none, equiv = FV2MAX(I)]

(I,5) Minimum surface area fraction for debris for component I.
[default = 0.0, units = none, equiv = FAMIN(I)]

(I,6) Maximum fraction of intact surface area blocked.
(default = 1.0 for CL, PD, and PB, I=2, 6, and 9, 0.0 otherwise; units = none, equiv = FBMAX)

(I,7) Surface-to-volume ratio for debris not in interstitial volume.

(default = 100.0, units = 1/m, equiv = SVRX)

1152 – Surface to Volume Ratio for Fluid

This coefficient is used to limit the surface area for heat transfer from COR to CVH when the core is blocked and the fluid volume is very small.

(1) Maximum surface-to-volume ratio for fluid, defining upper bound on heat transfer area for COR surfaces.
(default = 1000., units = m^{-1}, equiv = SVRFLU)

1200 – Smoothing of Heat Transfer Coefficients

The values of heat transfer coefficients calculated from appropriate correlations for heat transfer between the surfaces of COR components and the surrounding atmosphere and/or pool are averaged with the values from previous timesteps using

$$h_{new} = f_{old}\, h_{old} + (1 - f_{old})\, h_{calc}$$

to reduce numerical fluctuations. Different weightings of old and new are used for the atmosphere and the pool.

(1) Weight for old heat transfer to atmosphere.
(default = 0.5, units = none, equiv = none)

(2) Weight for old heat transfer to pool from quenched surfaces.
(default = 0.9, units = none, equiv = none)

(3) Weight for old heat transfer to pool from unquenched surfaces (not used in the current implementation where this coefficient is constant).
(default = 0.9, units = none, equiv = none)

1212 – Laminar Nusselt Numbers

These coefficients give the constant Nusselt number for various types of laminar forced convective flow.

(1) Nusselt number for fully developed flow in a circular tube with constant heat flux.
(default = 4.36, units = none, equiv = none)

(2) Nusselt number for fully developed flow in a rod bundle.
(default = 4.36, units = none, equiv = none)

1213 – Laminar Developing Flow

These coefficients are used to calculate a developing flow factor for laminar flow. The following equation gives the developing flow factor *g* in terms of the entrance length *z*, the equivalent diameter *D*, the Reynolds number Re, the Prandtl number Pr, and the sensitivity coefficients:

$$g = 1.0 + C1213\,(1)/([z/(D\,\mathrm{Re}\,\mathrm{Pr})] + C1213\,(2))$$

(1) Constant coefficient.
(default = 0.00826, units = none, equiv = none)

(2) Constant coefficient.
(default = 0.00110, units = none, equiv = none)

1214 – Turbulent Forced Convective Flow in Tubes

These coefficients are used to calculate the Nusselt number for forced convective flow in tubes, given by the following equation in terms of the Reynolds number Re and the Prandtl number Pr:

$$Nu = C1214(1)\mathrm{Re}^{C1214(2)}\,\mathrm{Pr}^{C1214(3)}$$

(1) Constant coefficient.
(default = 0.023, units = none, equiv = none)

(2) Reynolds number exponent.
(default = 0.8, units = none, equiv = none)

(3) Prandtl number exponent.
(default = 0.4, units = none, equiv = none)

1221 – Laminar Free Convection between Parallel Vertical Surfaces

These coefficients are used to calculate the Nusselt number for laminar free convection between parallel vertical surfaces. The following equation gives the Nusselt number Nu in terms of these coefficients and the Rayleigh number *Ra*, the height *L*, and the separation distance *D*:

$$Nu = C1221(1)Ra^{C1221(2)}(L/D)^{C1221(3)}$$

(1) Constant coefficient.
(default = 0.18, units = none, equiv = none)

(2) Rayleigh number exponent.
(default = 0.25, units = none, equiv = none)

(3) L/D exponent.
(default = -1./9., units = none, equiv = none)

1222 – Turbulent Free Convection between Parallel Vertical Surfaces

These coefficients are used to calculate the Nusselt number for turbulent free convection between parallel vertical surfaces. The following equation gives the Nusselt number *Nu* in terms of these coefficients and the Rayleigh number *Ra*, the height *L*, and the separation distance *D*:

$$Nu = C1222(1) Ra^{C1222(2)} (L/D)^{C1222(3)}$$

(1) Constant coefficient.
(default = 0.065, units = none, equiv = none)
(2) Rayleigh number exponent.
(default = 1./3., units = none, equiv = none)
(3) L/D exponent.
(default = -1./9., units = none, equiv = none)

1231 – Forced Convective Flow over a Spherical Particle

These coefficients are used to calculate the Nusselt number for forced convective flow over a single spherical particle. The following equation gives the Nusselt number *Nu* in terms of the Reynolds number Re and the Prandtl number Pr:

$$Nu = C1231(1) + C1231(2) \mathrm{Re}^{C1231(3)} \mathrm{Pr}^{C1231(4)}$$

(1) Additive constant (Stokes flow limit).
(default = 2.0, units = none, equiv = none)
(2) Constant coefficient.
(default = 0.60, units = none, equiv = none)
(3) Reynolds number exponent.
(default = 0.5, units = none, equiv = none)
(4) Prandtl number exponent.
(default = 1./3., units = none, equiv = none)

1232 – Free Convective Flow over a Spherical Particle

These coefficients are used to calculate the Nusselt number for free convective flow over a single spherical particle. The following equation gives the Nusselt number *Nu* in terms of the Grashof number *Gr* and the Prandtl number Pr:

$$Nu = C1232(1) + C1232(2) Gr^{C1232(3)} \Pr^{C1232(4)}$$

(1) Additive constant (Stokes flow limit).
(default = 2.0, units = none, equiv = none)

(2) Constant coefficient.
(default = 0.60, units = none, equiv = none)

(3) Grashof number exponent.
(default = 0.25, units = none, equiv = none)

(4) Prandtl number exponent.
(default = 1./3., units = none, equiv = none)

By default, boiling heat transfer in the COR package will be calculated using correlations from the HS package (see the HS Reference Manual). If C1241(5) is nonzero, the simplified correlations from the MARCH 2.0 code, specified by sensitivity coefficient arrays C1241 and C1242, will be used instead. In <u>either</u> case, the CHF temperature difference specified by C1241(4) will be used to determine whether debris in the core or lower head is quenched.

1241 – Simplified Nucleate Boiling Curve

If C1241(5) is nonzero, these coefficients are used to calculate a simplified nucleate boiling curve for pool boiling. The following equation gives the heat transfer coefficient *h* (W/m^2-K) in terms of the pressure *P* (Pa), the temperature difference ΔT (K), and these coefficients:

$$h = C1241(1) P^{C1241(2)} \Delta T^{C1241(3)}, \Delta T < C1241(4)$$

(1) Constant coefficient.
(default = 34.5, units = W/m^2-K-Pa$^{0.25}$-K$^{1.523}$, equiv = none)

(2) Pressure exponent.
(default = 0.25, units = none, equiv = none)

(3) Temperature difference exponent.
(default = 1.523, units = none, equiv = none)

(4) CHF temperature difference.
(default = 23.4, units = K, equiv = none)

(5) Option switch, nonzero to specify use of the simplified MARCH 2.0 model rather than correlations from the HS package.

(default = 0.0, units = none, equiv = none)

1242 – Simplified Transition Film Boiling Curve

If C1241(5) is nonzero, these coefficients are used to calculate a simplified transition boiling curve for pool boiling. The following equation gives the heat transfer coefficient *h* (W/m^2-K) in terms of the pressure P (Pa), the temperature difference ΔT (K), and these coefficients:

$$h = C1242(1) P^{C1242(2)} \Delta T^{C1242(3)}, \Delta T \geq C1241(4)$$

(1) Constant coefficient.
(default = 1.41E07, units = W/m^2-K-Pa$^{0.25}$-K$^{-2.575}$, equiv = none)

(2) Pressure exponent.
(default = 0.25, units = none, equiv = none)

(3) Temperature difference exponent.
(default = -2.575, units = none, equiv = none)

1244 – Debris Dryout Heat Flux Correlation

These coefficients are used to calculate the dryout heat flux for particulate debris beds using the zero-dimensional Lipinski turbulent correlation, given in Section 2.3.7 of the COR Package Reference Manual.

(1) Leading coefficient.
(default = 0.756, units = none, equiv = none)

(2) Capillary head for water and 0.5 mm particles.
(default = 0.089, units = m, equiv = ZCAP)

(3) Minimum debris porosity allowed.
(default = 0.15, units = none, equiv = none)

1245 – Downward-Facing Lower Head Heat Transfer Correlations

These coefficients are used to calculate downward heat transfer from the lower head to water in the reactor cavity. The critical heat flux as a function of the inclination of the surface ($\theta = 0^\circ$ for downward-facing surfaces) is given by

$$q_{CHF} = \left[C1245(1) + C1245(2) \cdot \theta^{C1245(3)}\right] \cdot (\rho_v)^{0.5} \cdot h_{lv} \cdot \left[g \cdot \sigma \cdot (\rho_l - \rho_v)\right]^{0.25}$$

where ρ_l and ρ_v are the densities of water and steam, respectively, σ is the interfacial surface tension between steam and water, g is the acceleration of gravity and h_{lv} is the latent heat of vaporization of water. The minimum heat flux at the beginning of stable flim boiling is given by

$$q_{MIN} = \left[C1245(4) + C1245(5) \cdot \theta^{C1234(6)}\right] \cdot (\rho_v)^{0.5} \cdot h_{lv} \cdot \left[g \cdot \sigma \cdot (\rho_l - \rho_v)\right]^{0.25}$$

and the flim-boiling heat transfer coefficient is given by

$$h_{FLM} = f(\theta) \cdot k_v \cdot \left[h_{lv} \cdot \rho_v \cdot g \cdot (\rho_l - \rho_v) / (\mu_v \cdot k_v \cdot \Delta T)\right]^{1/3}$$

where k_v is the thermal conductivity of steam, μ_v is the viscosity of steam, $\Delta T = T_{SURF} - T_{SAT}$ is the difference between the surface temperature and the saturation temperature, and $f(\theta)$ gives the angular dependence. There are two options for calculating $f(\theta)$. If C1245(7) = 0, then $f(\theta)$ is given by

$$f(\theta) = C1245(8) \cdot (\sin\theta)^{C1245(9)}$$

Otherwise, it is given by

$$f(\theta) = C1245(10) + C1245(11) \cdot \theta^{C1245(12)}$$

Details of the model are given in Section 6.1 of the COR Package Reference Manual.

(1) Additive constant in q_{CHF} correlation.
(default = 0.034, units = none, equiv = none)

(2) Multiplicative constant in q_{CHF} correlation.
(default = 0.0037, units = none, equiv = none)

(3) Exponent of θ in q_{CHF} correlation.
(default = 0.656, units = none, equiv = none)

(4) Additive constant in q_{MIN} correlation.
(default = 4.8E-4, units = none, equiv = none)

(5) Multiplicative constant in q_{MIN} correlation.
(default = 8.2E-4, units = none, equiv = none)

(6) Exponent of θ in q_{MIN} correlation.
(default = 0.407, units = none, equiv = none)

(7) Flim-boiling heat transfer coefficient correlation selection switch.
(default = 0., units = none, equiv = none)

(8) Multiplicative constant in default flim-boiling correlation.
(default = 0.142, units = none, equiv = none)

(9) Exponent of sin θ dependence in default flim-boiling correlation.
(default = 0.3333333, units = none, equiv = none)

(10) Additive constant in optional flim-boiling correlation.
(default = 0.055, units = none, equiv = none)

(11) Multiplicative constant in optional flim-boiling correlation.
(default = 0.016, units = none, equiv = none)

(12) Exponent of θ in optional flim-boiling correlation.
(default = 0.5, units = none, equiv = none)

1246 – Heat Transfer Coefficient, Lower Head to Atmosphere

This coefficient defines that heat transfer coefficient between the lower head and the atmosphere when the head is not completely covered by pool.

(1) Heat transfer coefficient to be used for that part of the lower head that is exposed to the atmosphere.
(default=10.0, units = W/m^2K, equiv = HATCAV)

1250 – Conduction Enhancement for Molten Components

These coefficients are used to enhance heat transfer at high temperatures, where core debris is molten, to capture the qualitative effects of convection in molten pools. This is done by using an enhanced thermal conductivity in the normal model for conduction between pairs of core components. The enhancement factor is given by

$$Factor = \max\left\{1.0, C1250(2)[T_{\max} - C1250(1)]^3\right\}$$

where $T_{\max}$ is the greater of the temperatures of the two components involved.

(1) Temperature above which enhancement is employed.
(default = 2800.0, units = K, equiv = TKMIN)

(2) Coefficient in enhancement.
(default = 0.01 units = K^{-1}, equiv = TKFAC)

1260 – Quench Model Parameters

The first three coefficients are used with the reflooding model in the correlation for quench front velocity. The fourth specifies the heat transfer from unquenched,

submerged surfaces. The quench velocity correlation implemented in the code is that of Dua and Tien, as recommended by Carbajo and Siegel. The velocity is given by

$$u = \alpha Pe / \delta ,$$

where

$$Pe = \left[\overline{B}\left(1 + 0.4\overline{B}\right)\right]^{1/2} ,\ \overline{B} = Bi(1-\Theta)^2 / \Theta ,\ Bi = \frac{C1260(3)\delta}{k} ,$$

and the temperature dependence is given by

$$\Theta = \frac{T_{surface} - T_{sat}}{C1260(1) - T_{sat}}$$

Here, δ is the surface thickness, α is the thermal diffusivity, and T_{sat} is the saturation temperature. The default value represents a best-estimate, based on the analysis of quench tests as discussed in Section 2.2.3 of the COR Package Reference Manual.

(1) Maximum excess temperature (superheating of the surface above saturation) against which a quench front can advance.
(default = 600.0, units = K, equiv = DTMAXQ)

(2) Minimum excess temperature (superheating of the surface above saturation) for a surface to be considered unquenched.
(default = 40.0, units = K, equiv = DTMINQ)

(3) Heat transfer coefficient for quenching. The only use is as h* in evaluation of the Biot Number in the quench velocity correlation.
(default = 1.5E+05, units = W/m^2K, equiv = HSTAR)

(4) Heat transfer coefficient for precursor cooling. Used with the unquenched surface temperature and the pool temperature for any unquenched portion of a surface that is below the two-phase mixture level.
(default = 125.0, units = W/m^2K , equiv = HTCPRE)

1270 – Flow Regime Map Parameters

These coefficients are used by a model that allows the COR package to infer a two-phase mixture level different from that calculated by the CVH package.

If there is (1) a significant volume of pool in a CVH control volume containing one or more core cells and (2) a nontrivial volume of pool in a second CVH control volume that

contains a core cell immediately above the first control volume, the current model infers the existence of a continuous (i.e., bridged) pool. That is, it assumes that all core cells contained in the first (lower) control volume are completely covered by the two-phase mixture. The current modeling is parametric and extremely simple.

(1) Minimum pool fraction in a control volume to consider bridging of the pool to an upper volume.
(default = 0.1, units = none, equiv = PLFRCT)

(2) Minimum pool fraction in a control volume to consider bridging of the pool from a lower volume.
(default = 0.001, units = none, equiv = PLFRMN)

1280 – Nusselt-Rayleigh Correlation for Molten Pool

These coefficients define the Nusselt-Rayleigh correlation for the convection in each molten pool. The first dimension indicates the heat transfer path, and the second dimension indicates the parameter 1 or 2. Heat transfer paths are in the following respective order: MP1 to lower head, MP1 to molten pool interface, MP1 to pool atmosphere boundary, MP2 to lower head, MP2 to molten pool interface, and MP2 to pool atmosphere boundary. The following example shows the correlation for Molten Pool 1 to the lower head:

$$Nusselt = C1280(1,1)Ra^{C1280(1,2)}$$

where *Nusselt* is the Nusselt number, and *Ra* is the Rayleigh number.

(1,1) Coefficient Constant.
(default = 0.3, units = N/A)

(1,2) Coefficient Constant
(default = 0.22, units = N/A)

1281 – Molten Pool Convection Inertial Time Constant

These coefficients define the inertial time constant for the convection of molten pools. The following equation serves as a factor to account for the time needed to change the convection currents over the timestep:

$$1 - e^{-t/C1281(1)}$$

where t is the timestep in seconds.

(1) Time Constant

(default = 0.1, units = seconds)

1290 – Molten Pool Convection Model Directional Parameters

These coefficients define the curve fit parameters for the molten pool convection model directional dependence used for the Nusselt number directional dependence as a function of polar angle. The directional dependence function is approximated for five H/R values. There are five sets of parameters *a-f* and *k*. The directional function is a piecewise function of the following:

$$f(\theta) = \begin{Bmatrix} a\theta^3 + b\theta^2 + c\theta \\ d\theta^2 + e\theta + f \end{Bmatrix} \begin{Bmatrix} 0 \le \theta \le \arccos(1 - k\,{}^{H}\!/_{R}) \\ \arccos(1 - k\,{}^{H}\!/_{R}) \le \theta \le \arccos(1 - {}^{H}\!/_{R}) \end{Bmatrix}$$

where H is the height of the molten pool, and R is the radius of the lower head.

The parameters *a*, *b*, *c*, *d*, *e*, *f*, and *k* correspond to the first dimension of C1290(i, j). The second dimension of C1290 corresponds to the five evaluated points of H/R (0, 0.25, 0.5, 0.75, and 1).

	a (i=1)	b (i=2)	c (i=3)	d (i=4)	e (i=5)	f (i=6)	k (i=7)
H/R = 0 (j=1)	4.31	0.292	-0.0313	-3.11	0.777	0.473	0.56
H/R = 0.25(j=2)	4.31	0.292	-0.0313	-3.11	0.777	0.473	0.6
H/R = 0.5(j=3)	1.29	0.333	-0.0528	-0.212	0.626	0.53	0.64
H/R = 0.75(j=4)	0.549	0.361	-0.0727	-0.181	0.561	0.564	0.67
H/R = 1(j=5)	0.242	0.385	-0.0928	-0.169	0.531	0.584	0.7

3.2 Fission and Decay Power Parameters

1301 – Chexal-Layman Fission Power/Liquid Level Correlation

These coefficients define the constants in the Chexal-Layman fission power/liquid level correlation (described in Section 2.6 of the COR Package Reference Manual), given by the equations:

$$QFRAC = C1301(1)(CUNIT1 \cdot H_r)^{C1301(3)}[(P / C1301(6)]^{C1301(2)}(H / H_r)^{C1301(3)}$$

And

$$H = \max(0.0, \{L + C1301(4)[P / C1301(6)]^{C1301(5)}\})$$

where *QFRAC* is the fraction of full operating power, *P* is the pressure, *L* is the height of the downcomer water relative to the top of active fuel, H_r is an arbitrary reference height

(selected as 1 m) inserted in the correlation to render it nondimensional in form, and *CUNIT1* is a dimensional constant embedded in the code equal to 3.28084 m^{-1}.

(1) Multiplicative constant.
(default = 3.7E 02, units = none, equiv = none)

(2) Pressure exponent.
(default = 0.3, units = none, equiv = none)

(3) Liquid level exponent.
(default = 0.7, units = none, equiv = none)

(4) Zero power liquid level.
(default = 2.4384, units = m, equiv = none)

(5) Pressure exponent for zero power liquid level.
(default = 0.45, units = none, equiv = none)

(6) Reference pressure.
(default = 7.65318E06, units = Pa, equiv = none)

1311 – Material Fission Power Absorption Efficiencies

These coefficients specify the relative absorption efficiencies of fission power by the core materials. The lumping of materials (Zr with ZrO_2, steel with steel oxide, and Inconel with Zr) used in previous versions of MELCOR has been eliminated. The coefficients are used in conjunction with coefficients in array 1312 to distribute the fraction 1.0-C1312(1) of the fission power that escapes the UO_2 fuel component (i.e., gammas) and are applied only to materials in active components, as defined by the remainder of sensitivity coefficient array 1312. The default values differ for BWRs and PWRs. Note that if values for PWRs are modified in MELGEN, the output will print BWR values as the old values.

(1) UO_2 efficiency.
[default = 0.735 BWR, or 0.5 PWR, units = none, equiv = FEFF(1)]

(2) Zr efficiency.
[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FEFF(2)]

(3) Steel efficiency.
[default = 0.292 BWR, or 0.565 PWR, units = none, equiv = FEFF(3)]

(4) Control poison efficiency.
[default = 0.263 BWR, or 0.234 PWR, units = none, equiv = FEFF(4)]

(5) ZrO_2 efficiency.

[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FEFF(5)]

(6) steel oxide efficiency.

[default = 0.292 BWR, or 0.565 PWR, units = none, equiv = FEFF(6)]

(7) Inconel efficiency.

[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FEFF(7)]

1312 – Component Fission Power Absorption Parameters

These coefficients define the fraction of fission power that escapes the fuel component and determine which of the various COR components (2 through 6) will absorb this available fission power, based on the relative absorption efficiencies in array 1311. The fraction of fission power that escapes the fuel component is given by the quantity [1 – POWFAC(1)]. If POWFAC(1) = 1.0, all fission power will be absorbed by the fuel, regardless of the material efficiencies specified in sensitivity coefficient array 1311. The remaining elements of the array, POWFAC(2:9), serve as flags to indicate the participation of the corresponding component. Only components with nonzero values are considered able to absorb fission power. Currently, energy will not be conserved, with messages generated, unless all values are zero or all are nonzero.

(1) UO_2 fuel initial absorption fraction.

[default = 0.9, units = none, equiv = POWFAC(1)]

(2) Participation flag for cladding.

[default = 1.0, units = none, equiv = POWFAC(2)]

(3) Participation flag for canister portion not adjacent to control blade.

[default = 1.0, units = none, equiv = POWFAC(3)]

(4) Participation flag for canister portion adjacent to control blade.

[default = 1.0, units = none, equiv = POWFAC(4)]

(5) Participation flag for former.

[default = 1.0, units = none, equiv = POWFAC(5)]

(6) Participation flag for particulate debris.

[default = 1.0, units = none, equiv = POWFAC(6)]

(7) Participation flag for supporting structure.

[default = 1.0, units = none, equiv = POWFAC(7)]

(8) Participation flag for nonsupporting structure.

[default = 1.0, units = none, equiv = POWFAC(8)]

(9) Participation flag for particulate debris in bypass.

[default = 1.0, units = none, equiv = POWFAC(9)]

1321 – Material Absorption Efficiencies for Decay Heat

These coefficients specify the relative absorption efficiencies of decay heat by the core materials. They are used in conjunction with coefficients in array 1322 to distribute the fraction 1.0-C1322(1) of the decay power that escapes the UO_2 fuel component (i.e., gammas), and are applied only to materials in active components, as defined by the remainder of sensitivity coefficient array 1312. The default values differ for BWRs and PWRs. Note that if values for PWRs are modified in MELGEN, the output will print BWR values as the old values. Defaults are based on ORNL report ORNL/NRC/LTR-94/42, which provides derivations and guidance for evaluation of nondefault values.

(1) UO_2 efficiency.
[default = 0.735 BWR, or 0.5 PWR, units = none, equiv = FDHEFF(1)]

(2) Zr efficiency.
[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FDHEFF(2)]

(3) Steel efficiency.
[default = 0.292 BWR, or 0.565 PWR, units = none, equiv = FDHEFF(3)]

(4) Control poison efficiency.
[default = 0.263 BWR, or 0.234 PWR, units = none, equiv = FDHEFF(4)]

(5) ZrO_2 efficiency.
[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FDHEFF(5)]

(6) Steel oxide efficiency.
[default = 0.292 BWR, or 0.565 PWR, units = none, equiv = FEFF(6)]

(7) Inconel efficiency.
[default = 0.4 BWR, or 0.541 PWR, units = none, equiv = FDHEFF(7)]

1322 – Component Decay Power Absorption Parameters

These coefficients define the fraction of decay power that escapes the fuel component, and determine which of the various COR components (2 through 6) will absorb this available decay power, based on the relative absorption efficiencies in array 1321. The fraction of decay power that escapes the fuel component is given by the quantity [1 – PWDHFC(1)]. If PWDHFC(1) = 1.0, all decay power will be absorbed by the fuel,

regardless of the material efficiencies specified in sensitivity coefficient array 1321. The remaining elements of the array, PWDHFC(2:9), serve as flags to indicate the participation of the corresponding component. Only components with nonzero values are considered able to absorb decay power. Currently, energy will not be conserved, with messages generated, unless all values are zero or all are nonzero.

(1) UO_2 fuel initial absorption fraction.
[default =0.5, units = none, equiv = PWDHFC(1)]

(2) Participation flag for cladding.
[default = 1.0, units = none, equiv = PWDHFC(2)]

(3) Participation flag for canister portion not adjacent to control blade.
[default = 1.0, units = none, equiv = PWDHFC(3)]

(4) Participation flag for canister portion adjacent to control blade.
[default = 1.0, units = none, equiv = PWDHFC(4)]

(5) Participation flag for former.
[default = 1.0, units = none, equiv = PWDHFC(5)]

(6) Participation flag for particulate debris.
[default = 1.0, units = none, equiv = PWDHFC(6)]

(7) Participation flag for supporting structure.
[default = 1.0, units = none, equiv = PWDHFC(7)]

(8) Participation flag for nonsupporting structure.
[default = 1.0, units = none, equiv = PWDHFC(8)]

(9) Participation flag for particulate debris in bypass.
[default = 1.0, units = none, equiv = PWDHFC(9)]

3.2.1 Numerical Control Parameters

1401 – Timestep Control Parameters

These coefficients are used to control the system timestep to prevent instabilities in the interface between the COR package and the CVH package. The interpretation of these coefficients is as follows:

(1) Maximum ratio by which the timestep for the next cycle may be increased.
(default = 1.6, units = none, equiv = DTINC)

(2) Minimum ratio by which the timestep for the next cycle may be decreased.
(default = 0.8, units = none, equiv = DTDEC)

(3) Minimum value of the ratio between current cycle energy transfer and previous cycle energy transfer before requiring a repeat of the current cycle.

(default = -1.0, units = none, equiv = none)

(4) Maximum value of the ratio between current cycle energy transfer and previous cycle energy transfer before requiring a repeat of the current cycle.

(default = 20.0, units = none, equiv = none)

(5) Maximum absolute difference from 1.0 of the ratio between current cycle energy transfer and previous cycle energy transfer for any core volume that will allow an increase in the next cycle timestep.

(default = 0.5, units = none, equiv = RATINC)

(6) Minimum absolute difference from 1.0 of the ratio between current cycle energy transfer and previous cycle energy transfer for any volume that will force a decrease in the next cycle timestep.

(default = 1.0, units = none, equiv = RATDEC)

1402 – Candling Control Parameter

This coefficient controls the fraction of a material in a component that must be molten before it is considered available to candle. The default value serves only to avoid numerical problems.

(1) Threshold melt fraction for candling of a material in a component.

(default = 10 times unit roundoff, units = none, equiv = EPSCND)

1403 – Point Kinetics Time-Step Control

This coefficient is the fraction of the calculated reactor time-period to be used as maximum time-step advancement for the point kinetics model.

(1) Fraction of reactor time-period to be used as maximum time-step.

(default = 1.0, units = none, equiv = PKTSC)

1404 – Temperature Feedback Reactivity Coefficients

These coefficients define the curve-fit for the temperature feedback reactivity coefficients for core Doppler, fuel expansion, and moderator expansion for the point kinetics model.

(1) Core Doppler temperature reactivity coefficient.
(default = -2.2E-2, units = none, equiv = RK_D)

(2) Fuel expansion temperature reactivity coefficient.
(default = -4.78E-5, units = K^{-1}, equiv = RK_F1)

(3) Fuel expansion temperature reactivity coefficient.
(default = 6.75E-9, units = K^{-2}, equiv = RK_F2)

(4) Graphite expansion temperature reactivity coefficient.
(default = 14.834E-5, units = K^{-1}, equiv = RK_G1)

(5) Graphite expansion temperature reactivity coefficient.
(default = -1.6025E-7, units = K^{-2}, equiv = RK_G2)

(6) Graphite expansion temperature reactivity coefficient.
(default = 6.991E-11, units = K^{-3}, equiv = RK_G3)

(7) Graphite expansion temperature reactivity coefficient.
(default = -1.114E-14, units = K^{-4}, equiv = RK_G4)

1405 – Point Kinetics Model 6 Group Parameters

These coefficients define the nuclear physics data for the 6-point kinetics model.

(1,1) Decay constant for first delayed-neutron group.
(default = 1.24E-2, units = $seconds^{-1}$, equiv = LAMBDA(1))

(2,1) Decay constant for second delayed-neutron group.
(default = 3.05E-2, units = $seconds^{-1}$, equiv = LAMBDA(2))

(3,1) Decay constant for third delayed-neutron group.
(default = 1.11E-1, units = $seconds^{-1}$, equiv = LAMBDA(3))

(4,1) Decay constant for fourth delayed-neutron group.
(default = 3.01E-1, units = $seconds^{-1}$, equiv = LAMBDA(4))

(5,1) Decay constant for fifth delayed-neutron group.
(default = 1.14, units = $seconds^{-1}$, equiv = LAMBDA(5))

(6,1) Decay constant for sixth delayed-neutron group.
(default = 3.01, units = $seconds^{-1}$, equiv = LAMBDA(6))

(1,2) Relative abundance of first delayed-neutron group.
(default = 3.3E-2, units = none, equiv = F(1))

(2,2) Relative abundance of second delayed-neutron group.
(default = 2.19E-1, units = none, equiv = F(2))

(3,2) Relative abundance of third delayed-neutron group.
(default = 1.96E-1, units = none, equiv = F(3))

(4,2) Relative abundance of fourth delayed-neutron group.
(default = 3.95E-1, units = none, equiv = F(4))

(5,2) Relative abundance of fifth delayed-neutron group.
(default = 1.15E-1, units = none, equiv = F(5))

(6,2) Relative abundance of sixth delayed-neutron group.
(default = 4.2E-2, units = none, equiv = F(6))

1406 – Point Kinetics Nuclear Data

These coefficients define the nuclear physics data for the point kinetics model.

(1) Prompt neutron generation time.
(default = 1E-2, units = seconds, equiv = XLPROMPT)

(2) Total effective delayed-neutron yield.
(default = 6.5E-3, units = none, equiv = BETA_SUM)

(3) Average number of neutrons produced per thermal fission.
(default = 2.432, units = #, equiv = NU)

(4) Immediate (prompt and delayed) energy released per fission.
(default = 3.044E-11, units = J, equiv =QF)

(5) Average delayed neutron lifetime.
(default = 8.5E-2, units = seconds, equiv =LEFF)

(6) Prompt neutron lifetime.
(default = 1E-4, units = seconds, equiv = LP)

3.2.2 Geometric Parameters

1501 – Canister Mass/Surface Area Splits

These coefficients specify the fractions of the input values for canister mass and surface area that are assigned to the two canister components (one adjacent to the control blade, the other not). They also specify the split between inner and outer (bypass and channel) surfaces.

(1) Fraction of canister mass in component adjacent to control blade.
(default = 0.5, units = none, equiv = FMCB)

(2) Fraction of canister surface area on inner (fuel-side) surface of component adjacent to the control blade.
(default = 0.5, units = none, equiv = FACBC)

(3) Fraction of canister surface area on outer (bypass-side) surface of component adjacent to control blade.
(default = 0.5, units = none, equiv = FACBB)

(4) Fraction of canister surface area on inner (fuel-side) surface of component not adjacent to the control blade.
(default = 0.5, units = none, equiv = FACNC)

(5) Fraction of canister surface area on outer (bypass-side) surface of component not adjacent to the control blade.
(default = 0.5, units = none, equiv = FACNB)

1502 – Minimum Component Masses

These coefficients specify the minimum component mass below which the masses and energies will be discarded and the minimum component mass below which the component will not be subject to the maximum temperature change criterion. The default value of C1502(2) must be reduced to successfully simulate small-scale experiments.

(1) Minimum total mass of component.
(default = 1.0E-6, units = kg, equiv = XMCMN1)

(2) Minimum total mass of component subject to the maximum temperature change criterion for timestep control.
(default = 10.0, units = kg, equiv = XMCMN2)

1503 – Core Blockage Parameters

This sensitivity coefficient array has been replaced by array 1505. It is no longer used, and input for it is treated as a fatal error

1505 – Core Blockage Parameters

These coefficients specify the geometric parameters affecting core flow resistance and heat transfer under conditions of flow blockage. Unlike coefficient 1503, which it replaces, distinct values are used in the calculation of flow blockage and of heat transfer areas.

(1) Minimum porosity to be used in calculating the flow resistance in the flow blockage model.

(default = 0.05, units = none, equiv = PMNBLK)

(2) Minimum porosity to be used in calculating the area for heat transfer to fluid.

(default = 0.05, units = none, equiv = PMNSRF)

1504 – Core Cell Volume Consistency Tolerances

These coefficients specify the tolerances on internal consistency in representation of volumes within the COR package database. If either of the limits is exceeded in any core cell, a message will be issued (limited to 100 times each in any execution). If the discrepancy in total volume exceeds 100 times TOLVT, the calculation will be terminated.

(1) Permitted relative error in total volume in any cell.

(default = ten times unit roundoff, units = none, equiv = TOLVT)

(2) Permitted relative error in channel or bypass portion of cell volume in a BWR.

(default = 1.0E-4, units = none, equiv = TOLV1)

3.2.3 Lower-Head Mechanical Model Parameters

1600 – Model Parameters

These coefficients specify the desired level of modeling detail.

(1) Switch to choose zero-dimensional (0.0) or one-dimensional (1.0) modeling of the stress and strain distribution in the lower head.

(default = 1.0, units = none, equiv = none)

(2) Linear expansivity of the lower-head-load-bearing material. (Note, this value is only used if the one-dimensional model has been selected.)

(default = 1.E-5, units = K^{-1}, equiv = none)

(3) Differential pressure lower limit. The mechanical model will be bypassed whenever the effective differential pressure across the lower head falls below this value. Hence, a large value for this coefficient (e.g., 1.E10) will totally disable the mechanical model.

(default = 1.E3, units = Pa, equiv = none)

1601 – Larson-Miller Creep-Rupture Parameters for Vessel Steel

The Larson-Miller parameter for vessel steel is given by

$$P_{LM} = C1601(1) \cdot \log_{10} \sigma_e + C1601(2)$$

where σ_e is the effective stress in Pa, and the coefficients are material dependent. The default values are typical for reactor vessel carbon steel. The time to failure (rupture) in seconds is given by

$$t_R = 10^{[P/T - C1601(3)]}$$

where T is the temperature in K, and the plastic strain at time $t + \Delta t$ is given by

$$\varepsilon_{pl}(t + \Delta t) = \varepsilon_{pl}(t) + C1601(4) \cdot \Delta t / T_R$$

The default parameters used in the Larson-Miller Creep Rupture model for failure of the lower head have been modified. The values now used are those found for SA533B1 vessel steel in the SNL Lower Head Failure Program (NUREG/CR5582):

(1) Inherently negative multiplicative constant.
(default = -4.725E3, units = none, equiv = none)

(2) Inherently positive additive constant.
(default = 4.812E4, units = none, equiv = none)

(3) Additive exponential constant.
(default = 7.042, units = none, equiv = none)

(4) Total strain assumed to cause failure.
(default = 0.18, units = none, equiv = none)

1602 – Vessel Steel Elastic Modulus Parameters

The elastic modulus of vessel steel is given as a function of temperature, T, by

$$E(T) = C1602(1) \cdot \left(\left\{ [1 + [T / C1602(3)]^{C1602(4)} \right\}^{-1} - \left\{ 1 + [C1602(2) / C1602(3)]^{C1602(4)} \right\}^{-1} \right)$$

(1) Leading multiplicative constant.
(default = 2.E11, units = Pa, equiv = none)

(2) Temperature at which elastic modulus vanishes.
(default = 1800.0, units = K, equiv = none)

(3) Temperature at which elastic modulus is approximately halved.
(default = 900.0, units = K, equiv = none)

(4) Exponent of scaled temperatures.
(default = 6.0, units = none, equiv = none)

1603 – Vessel Steel Yield Stress Parameters

The yield stress of vessel steel is given as a function of temperature, *T*, by

$$\sigma_Y(T) = C1603(1) \cdot \left(\left\{ [1 + [T / C1603(3)]^{C1603(4)} \right\}^{-1} - \left\{ 1 + [C1603(2) / C1603(3)]^{C1603(4)} \right\}^{-1} \right)$$

(1) Leading multiplicative constant.
(default = 4.E8, units = Pa, equiv = none)

(2) Temperature at which yield stress vanishes.
(default = 1700.0, units = K, equiv = none)

(3) Temperature at which yield stress is approximately halved.
(default = 900.0, units = K, equiv = none)

(4) Exponent of scaled temperatures.
(default = 6.0, units = none, equiv = none)

3.2.4 Support Structure Mechanical Model Parameters

1604 – Larson-Miller Creep-Rupture Parameters for Support Structures

The Larson-Miller parameter for steel in support structures is given by

$$P_{LM} = C1604(1) \cdot \log_{10} \sigma_e + C1604(2)$$

where σ_e is the effective stress in Pa, and the coefficients are material dependent. The default values are typical for stainless steels. The time to failure (rupture) in seconds is given by

$$t_R = 10^{[P/T - C1604(3)]}$$

where *T* is the temperature in K.

(1) Inherently negative multiplicative constant.
(default = -7.500E3, units = none, equiv = none)

(2) Inherently positive additive constant.
(default = 8.100E4, units = none, equiv = none)

(3) Additive exponential constant.
(default = 16.44, units = none, equiv = none)

1605 – Internal Steel Elastic Modulus Parameters

The elastic modulus of steel in support structures is given as a function of temperature, *T*, by

$$E(T) = C1605(1) \cdot \left(\left\{ [1 + [T / C1605(3)]^{C1605(4)} \right\}^{-1} - \left\{ 1 + [C1605(2) / C1605(3)]^{C1605(4)} \right\}^{-1} \right)$$

(1) Leading multiplicative constant.
(default = 370.E9, units = Pa, equiv = none)

(2) Temperature at which elastic modulus vanishes.
(default = 1700., units = K, equiv = none)

(3) Temperature at which elastic modulus is approximately halved.
(default = 1650., units = K, equiv = none)

(4) Exponent of scaled temperatures.
(default = 3.0, units = none, equiv = none)

1606 – Internal Steel Yield Stress Parameters

The yield stress of steel in support structures is given as a function of temperature, *T*, by

$$\sigma_Y(T) = C1606(1) \cdot \left(\left\{ [1 + [T / C1606(3)]^{C1606(4)} \right\}^{-1} - \left\{ 1 + [C1606(2) / C1606(3)]^{C1606(4)} \right\}^{-1} \right)$$

(1) Leading multiplicative constant.
(default = 260.E6, units = Pa, equiv = none)

(2) Temperature at which yield stress vanishes.
(default = 1700., units = K, equiv = none)

(3) Temperature at which yield stress is approximately halved.
(default = 800., units = K, equiv = none)

(4) Exponent of scaled temperatures.
(default = 3.0, units = none, equiv = none)

4 Plot Variables

The plot variables currently included in the COR package are listed below, along with a brief description. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

Several variables are available for each COR component. Available COR components include FU (fuel), CL (clad), CN (canister), SS (support structure), NS (non-support

structure), SH (shroud), RK (SFP rack), FM (former), MP1 (oxide molten pool in channel), MB1 (oxide molten pool in bypass), MP2 (metallic molten pool in channel), and MB2 (metallic molten pool in bypass). Interpretation of each component is given by the table in Section 1.1.

To reduce the size of the file, each COR package plot variable can be enabled or disabled in the binary plot file. The COR_PLOTS table provides the interface to the user for enabling or disabling variables from the plot file, where each record provides a reference to a plot variable followed by a field indicating whether the variable is optionally included by default in the plot file ('ON') or optionally not included by default ('OFF').

```
COR_PLOTS  3
1  COR-TCMP  OFF
2  COR-HTC-P  ON
3  COR-ZQ  OFF
```

The reference to each plot variable is generally the plotkey for the plot variable. In some cases, several plot variables may be enabled/disabled by a single reference. For those plotvariables, the appropriate reference is provided below with the definition. The default treatment for enabling/disabling the plot variable is also shown for each plot variable. If the default is 'ALWAYS' this indicates that the plot variable is not optional and will always be included in the plotfile.

4.1 Core Mass Variables

COR-M<MAT>.n	Total mass of material MAT in cell n. The choices allowed for MAT are CRP, INC, SS, SSOX, UO2, ZR, and ZRO2. (units = kg, default = ON, Reference = 'COR-MTOT')
COR-M<MAT>-TOT	Total mass of material MAT in core. The choices allowed for MAT are CRP, INC, SS, SSOX, UO2, ZR, and ZRO2. (units = kg, default = ON, Reference = 'COR-MMAT-TOT')

In the following four variables, acceptable strings for material are UO2, ZR, SS, INC, ZRO2, SSOX, and CRP. (Interpretation is as in previous variables.)

COR-DC.m.k.n	Conglomerate debris mass for material m on component k in cell n. (units = kg, default = ON)
COR-PB.m.n	Particulate debris mass for material m in bypass of cell n.

(units = kg, default = ON)

COR-PD.m.n — Particulate debris mass for material m in cell n.
(units = kg, default = ON)

COR-M.M.m.k.n — Total mass (intact plus conglomerate) of material m in component k in cell n. If FU contains optional, additional materials, their masses are available through this variable.
(units = kg, default = ON, Reference= 'COR-M-MAT-COMP')

Examples

```
COR-DC.UO2.FU.201
COR-PB.UO2.201
COR-PD.UO2.201
COR-M.MUO2.FU.201
```

4.2 Core Flow Area and Component Volume Variables

COR-VOLF-<COMP>.n — Volume fraction of component COMP in cell n. In addition to the standard component set described above, the following are also available as pseuto components: FL (total fluid), FLB (fluid in bypass), FLC (fluid in channel), POR (intersticial porosity in channel PD), and PORB (intersticial porosity in channel).
(units = none, default = ON, Reference= 'COR-M-MAT-COMP')

Examples

```
COR-VOLF-SS.305
COR-VOLF-MP1.201
COR-VOLF-FM.304
```

4.3 Core Temperature Variables

COR-T< COMP >.n — Temperature of component 'COMP' in cell n where the choices allowed for COMP are FU (fuel), CL (clad), CN (canister), SS (support structure), NS (non-support structure), SH (shroud), FM (former), RK (SFP rack), MP1 (oxide molten pool in channel), MB1 (oxide molten pool in bypass), MP2 (metallic molten pool in channel), MB2 (metallic molten pool in bypass),

(units = K, default = ON, Reference = 'COR-TCMP')

COR-TSVC.n — Local channel fluid temperature seen by cell n.
(units = K, default = ON)

COR-TSVB.n — Local bypass fluid temperature seen by cell n.
(units = K, default = ON)

Examples

```
COR-TSS.305
COR-TMP1.201
COR-TFM.304
COR-TSVC.208
```

4.4 Core Energy and Heat Transfer Variables

COR-ENERGY-TOT — Total internal energy in COR package.
(units = J, default = ON)

COR-EFPD-TOT — Total cumulative fission power and decay heat generated in core.
(units = J, default = ON)

COR-EMWR-TOT — Total cumulative oxidation heat generated in core.
(units = J, default = ON)

COR-EB4C-TOT — Total cumulative heat from B_4C reactions in core.
(units = J, default = ON)

COR-ECNV-TOT — Total cumulative energy transfer to CVH package.
(units = J, default = ON)

COR-EBND-TOT — Total cumulative energy transfer to HS package.
(units = J, default = ON)

COR-EFPD-RAT — Total fission power and decay heat generation rate in core.
(units = W, default = ON)

COR-EMWR-RAT — Total oxidation heat generation rate in core.
(units = W, default = ON)

COR-EB4C-RAT	Total heat generation rate from B_4C reactions in core. (units = W, default = ON)
COR-ECNV-RAT	Total convective heat generation rate in core. (units = W, default = ON)
COR-EBND-RAT	Total radiative heat transfer rate to boundary heat structures. (units = W, default = ON)
COR-QCNV.m	Heat transfer rate (including radiation) to atmosphere in fluid volume m. (units = W, default = ON)
COR-HTCLH.i	Heat transfer coefficient from segment i of lower head to water pool. (units = W/m^2K, default = ON)
COR-HTCLH-AVE	Average heat transfer coefficient from lower head to water pool. (units = W/m^2K, default = ON)
COR-QFLXLH.i	Heat flux from segment i of lower head to water pool. (units = W/m^2, default = ON)
COR-QFLXLH-AVE	Average heat flux from lower head to water pool. (units = W/m^2)
COR-QTOTLH.i	Cumulative heat transferred from segment i of lower head to water pool. (units = J, default = ON)
COR-QTOTLH-TOT	Total cumulative heat transferred from lower head to water pool. (units = J, default = ON)
COR-ELHPOW.n	Electrical heating power generated in cor cell n (for calculations using electrical heating power option). (units = W, default = OFF)
COR-EM.k.n	Local emissivity of component k in cell n. (units = dimensionless, default = OFF)

COR-HTC-A.k.n	Local heat transfer coefficient to atmosphere for component k in cell n. (units = J/m-K, default = OFF)
COR-HTC-P.k.n	Local heat transfer coefficient to pool for component k in cell n. (units = J/m-K, default = OFF)
COR-HTC-S.k.n	Local heat transfer coefficient to pool surface for component k in cell n. (units = J/m-K, default = OFF)
COR-QCNV.k.n	Local convective heat addition from component k in COR cell n. (units = W/kg, default = OFF)
COR-DHEAT.k.n	Local decay heat addition from component k in COR cell n. (units = W/kg, default = OFF)
COR-QCND..k.n	Local conductive heat addition from component k in COR cell n. (units = W/kg, default = OFF)
COR-POW-Z.n	Local linear power density in cell n. (units = W/m, default = OFF)

4.5 Core Structural Loading and Damage Variables

COR-SS-STRESS.ijj	Maximum component of stress in SS in core cell ijj. (units = Pa, default = ON)
COR-SS-DAMAGE.ijj	Damage fraction (fraction of damage that will produce failure) for SS in core cell ijj. (units = none, default = ON)
COR-SS-TLEFT.ijj	Remaining life for SS in core cell ijj (for unchanged temperature and load). (units = s, default = ON)

4.6 Core Oxidation and B_4C Reaction Mass Variables

COR-DMH2-TOT	Total cumulative hydrogen production in core from all oxidation processes. (units = kg, default = ON)

COR-DMH2-SS	Total cumulative hydrogen production by oxidation of stainless steel in core. (units = kg, default = ON)
COR-DMH2-ZIRC	Total cumulative hydrogen production by oxidation of Zircaloy in core. (units = kg, default = ON)
COR-DMH2-B4C	Total cumulative hydrogen production by oxidation of B_4C in core. (units = kg, default = ON)
COR-DMCO-TOT	Total cumulative carbon monoxide production in core. (units = kg, default = ON)
COR-DMCO2-TOT	Total cumulative carbon dioxide production in core (available only for MCRP = B4C). (units = kg, default = ON)
COR-DMCH4-TOT	Total cumulative methane production in core (available only for MCRP = B4C). (units = kg, default = ON)
COR-ZROX-TLEFT.n	Time left in lifetime for clad component in cell n. (units = sec, default = ON)
COR-OXTH-<COMP>.n	Local oxide thickness for component 'COMP' in cell n. (units = m, default = OFF)
COR-H2T.n	Total local hydrogen generation for all components in cell n (intact and conglomerate). (units = kg, default = OFF)
COR-H2CT.n	Local hydrogen generation for all components in cell n from conglomerate masses. (units = kg, default = OFF)
COR-H2-<COMP>.n	Local hydrogen generation for component <COMP> in cell n (intact and conglomerate). (units = kg, default = OFF)
COR-H2C-<COMP>.n	Local hydrogen generation for component <COMP> in cell n from oxidation of the conglomerate mass.

	(units = kg, default = OFF)
COR-DISTC.n,ng	Local channel conglomerate mass in the sub-grid associated with cell n, in elevation ng (numbered from bottom of COR cell) in cell n. (units = kg, default = OFF)
COR-DISTB.n,ng	Local bypass conglomerate mass in the sub-grid associated with cell n, in elevation ng (numbered from bottom of COR cell). (units = kg, default = OFF)
COR-DCZ.m.k.n	Local linear mass density of conglomerated for material m, in component k, in COR cell n. (units = kg/m, default = OFF)
COR-CL-TH	Local metallic thickness for the clad component in cell n. (units = m, default = OFF)
COR-CRUST-THICK.irs	Local thickness of crust adjacent to lower head for segment irs. (units = m, default = OFF)

4.7 Core Molten Pool Variables

Masses

COR-M-<P>-<MP>	Mass of the convecting oxide molten pool <MP=MP1> or the metallic molten pool <MP=MP2> in the active COR <P=UP> or in the lower plenum<P=LP>. (units = kg, default = ON)

Temperatures

COR-T-<MP>-<P>	Temperature of the convecting oxide molten pool <MP=MP1> or the metallic molten pool <MP=MP2> in the active COR <POOL=UP> or in the lower plenum<POOL=LP>. (units = K, default = ON)

Volumes

COR-V-<MP>-<P>

Volume of the convecting oxide molten pool <MP=MP1> or the metallic molten pool <MP=MP2> in the active COR <POOL=UP> or in the lower plenum<POOL=LP>.

(units = m^3, default = ON)

Rayleigh Number

COR-RA1-<MP>-<P>

Rayleigh of the convecting oxide molten pool <MP=MP1> or the metallic molten pool <MP=MP2> in the active COR <POOL=UP> or in the lower plenum<POOL=LP>.

(units = -, default = ON)

```
Example:
COR-RA1-MP1-COR
```

Energy Balance Upper Pool

COR-RAD-<MP>-<P>

Radiation power from upper surface of pool where <MPX> can be MP1 (oxide) or MP2 (metallic) and <P> can be in the active core <COR> or in the lower plenum <LP>

(units = w, default = OFF)

```
Example:
COR-RAD-MP1-COR
```

COR-CONV-<MPX>-<P>

Convective heat removal rate from pool where <MPX> can be MP1 (oxide) or MP2 (metallic) and <P> can be in the active core <COR> or in the lower plenum <LP>

(units = w, default = OFF)

COR-QINT-<MPX>-<P>

Convective heat transfer at interface between stratified molten pools where <MPX> can be MP1 (oxide) or MP2 (metallic) and <P> can be in the active core <COR> or in the lower plenum <LP>

(units = w, default = OFF)

COR-QDCH-<MPX>-<P>

Decay heat power generated in pool where <MPX> can be MP1 (oxide) or MP2 (metallic) and <P> can be in the active core <COR> or in the lower plenum <LP>

(units = w, default = OFF)

4.8 Core Penetration, Lower Head, Breach and Melt Ejection Variables

COR-TPN.n	Temperature of penetration n. (units = K, default = ON)

In the following three variables, ijj refers to node jj of segment i, where jj = 01 is on the outside of the lower head. (Although nodal information on COR_LHN record is entered from inside to outside, the nodes are actually numbered in the opposite direction.)

COR-TLH.ijj	Temperature of vessel lower head node. (units = K, default = ON)
COR-VSTRAIN.ijj	Plastic strain in vessel lower head node. (units = dimensionless, default = ON)
COR-VSTRESS.ijj	Stress in vessel lower head node. (units = Pa, default = ON)
COR-MEJEC-TOT	Total debris mass ejected through vessel breach. (units = kg, default = ON)
COR-CAVJEC-<MAT>	Total mass ejected through vessel breach by material, MAT=(UO2, ZR, ZRO2, SS, or SSOX) (units = kg, default = OFF)

4.9 Core Accounting Variables

COR-DT	COR package timestep. (units = s, default = ALWAYS)
COR-NCYCLE	Total number of COR package subcycles. (units = none, default = ALWAYS)
COR-CPU	Total CPU usage by run portion of COR package. (units = s, default = ALWAYS)
COR-MASSERR	Total mass error for COR package. (units = kg, default = ALWAYS)
COR-ENERGYERR	Total energy error for COR package. (units = J, default = ALWAYS)
COR-REL-ENGY-ERR	COR relative energy error, defined as COR-ENERGYERR/ COR-ENERGY-TOT. (units = dimensionless, default = ALWAYS)

COR-REL-ENGY-ERM	COR relative energy error, modified to include enthalpy of mass transfers to BH and HS, defined as COR-ENERGYERR/(COR-ENERGY-TOT + debris to CAV). (units = dimensionless, default = ALWAYS)

5 Control Function Arguments

The control function arguments currently included in the COR package are listed below, along with a brief description. Note that plot variables—some are identical in definition to these control function arguments but different in format—are described in the previous section.

The choices allowed for SURF are:

FU	UO2 fuel
CL	Zircaloy cladding
CN	BWR canister (the part not adjacent to the control blade) surface adjacent to channel
CB	BWR canister-b (the part adjacent to the control blade) surface adjacent to channel
CNB	BWR canister (the part not adjacent to the control blade) surface adjacent to bypass
CBB	BWR canister-b (the part adjacent to the control blade) surface adjacent to bypass
SS	supporting structure
NS	nonsupporting structure
PB	particulate debris in bypass
PD	particulate debris in channel
SH	PWR core shroud -surface adjacent to channel
SHB	PWR core shroud – surface adjacent to bypass
FM	PWR core former
MP1	molten pool 1
MB1	molten pool 1 in bypass
MP2	molten pool 2
MB2	molten pool 2 in bypass
RK	spent fuel pool rack temperature

5.1 Core Mass Variables

COR-CellMass(IA,IR,MAT)	Total mass of material MAT in cell (IA,IR). The choices allowed for MAT are: CRP — Control Poison INC — Inconel SS — Stainless Steel SSOX — Stainless Steel Oxide

UO2	Uranium Dioxide
ZR	Zircaloy
ZRO2	Zirconium Dioxide
GRP	Graphite

(units = kg)

COR-TotMass(MAT) — Total mass of material MAT in core. The choices allowed for MAT are CRP, INC, SS, SSOX, UO2, ZR, and ZRO2. (Interpretation is as in the previous variable.)

(units = kg)

COR-CellMassSS(IA,IR,MAT) — Intact supporting structure (SS) MAT mass in cell (IA,IR). The choices allowed for MAT are SS, SSOX, ZR, ZRO2, and GRP.

(units = kg)

COR-CellMassNS(Ia,IR,MAT) — Intact nonsupporting structure (NS) MAT mass in cell (IA,IR). The choices allowed for MAT are SS, SSOX, ZR, ZRO2, and CRP.

(units = kg)

COR-CellMassSH(IA,IR,MAT) — Intact PWR core shroud (SH) MAT mass in cell (IA,IR). The choices allowed for MAT are SS, and SSOX.

(units = kg)

COR-CellMassFM(IA,IR,MAT) — Intact PWR core former (FM) MAT mass in cell (IA,IR). Only one choice is allowed for MAT: SS.

(units = kg)

In the following four variables, acceptable strings for material are UO2, ZR, SS, INC, ZRO2, SSOX, and CRP. (Interpretation is as in previous variables.) Acceptable strings for components are FU, CL, CN, CB, SS, NS, SH, FM, PB, PD, IMP1, IMP2, IMB1, and IMB2. (Interpretation is given by the table in Section 1.1.) For PMR, and PBR reactor types, MAT for COR-CellMass, and COR-CellMassSS includes GRP for graphite. These four control function arguments are always active, but the corresponding plot variables are active <u>only if IPMV has been set to 1 on MELCOR input record COR_EDV.</u>

COR-DC(IA,IR,COMP,MAT) — Conglomerate debris mass for material MAT on component COMP in cell (IA,IR).

(units = kg)

COR-PB(IA,IR,MAT) — Particulate debris mass for material MAT in bypass of cell (IA,IR).

(units = kg)

COR-PD(IA,IR,MAT)	Particulate debris mass for material MAT in cell (IA,IR). (units = kg)
COR-M(IA,IR,COMP,MAT)	Total mass (intact plus conglomerate) of material MAT in component COMP in cell (IA,IR). If FU contains optional, additional materials, their masses are available through this variable. (units = kg)
COR-CellMassFU(IA,IR,MAT)	Intact fuel MAT mass in cell (IA,IR). Only one choice is allowed for MAT: UO2. (units = kg)
COR-CellMassCL(IA,IR,MAT)	Intact cladding MAT mass in cell (IA,IR. The choices allowed for MAT are ZR and ZRO2. (units = kg)
COR-CellMassCN(IA,IR,MAT)	Intact canister (including CB) MAT mass in cell (IA,IR). The choices allowed for MAT are ZR and ZRO2. (units = kg)

In the following variable, debris includes particulate and conglomerate.

COR-CellMassDB(IA,IR,MAT)	Total debris mass of material MAT in cell (IA,IR). The choices allowed for MAT are SS, SSOX, UO2, ZR, ZRO2, and CRP. (units = kg)

5.2 Core Flow Area and Component Volume Variables

COR-AFLMIN(IA1,IA2,IR)	Minimum axial flow area open in core cells between (IA1,IR) and (IA2,IR), inclusive (channel and bypass combined for BWR). (units = m^2)
COR-VOL-FLU(IA,IR)	Volume of fluid (empty volume) in cell (IA,IR) (total for BWR). (units = m^3)
COR-VOL-FLUB(IA,IR)	Volume of fluid in bypass of cell (IA,IR) (zero for PWR). (units = m^3)

COR-VOL-FLUC(IA,IR) — Volume of fluid in channel of cell (IA,IR) (total for PWR).

(units = m^3)

COR-VOLF (IA,IR,COMP) — Volume fraction of component IC in cell (IA,IR). Note that COR-CellVolFrac and COR-VOLFRAC are also recognized. The choices allowed for COMP are FU, CL, CN, SS, NS, SH, FM, MP1, MB1, MP2, MB2 (interpretation is given by the table in Section 1.1.) and:

FL	Fluid volume fraction in cell
FLB	Fluid volume fraction in cell bypass of cell
FLC	Fluid volume fraction in channel of cell (IA,IR) (BWR only)
RK	Spent fuel pool rack
POR	Particulate debris porosity (interstitial) volume fraction in cell
PORB	Particulate debris porosity (interstitial) volume fraction in bypass cell

(units = none)

5.3 Core Temperature and Melt Fraction Variables

COR-CellTemp (IA,IR,COMP) — Temperature of COMP in cell (IA,IR). The choices allowed for COMP are:

FU	UO_2 fuel
CL	Zircaloy cladding
CN	BWR canister (the part not adjacent to the control blade)
CB	BWR canister-b (the part adjacent to the control blade)
SS	supporting structure
NS	nonsupporting structure
PB	particulate debris in bypass
PD	particulate debris in channel
SH	PWR core shroud
FM	PWR core former
MP1	molten pool 1
MB1	molten pool 1 in bypass
MP2	molten pool 2
MB2	molten pool 2 in bypass
SVC	local channel fluid temperature

	seen by cell SVB local bypass fluid temperature seen by cell RK spent fuel pool rack temperature (units = K)
COR-MLTFR(IA,IR,COMP,m)	Melt fraction of material number m in component COMP in cell (IA,IR). units = none)

5.4 Core Energy and Heat Transfer Variables

A number of variables of interest in energy accounting had previously been accessible only as plot variables. These were made accessible as control function arguments. Two new control function arguments were added to provide access to radiative heat transfer rates to boundary heat structures.

COR-ENERGY-TOT	Total internal energy in COR package. (units = J)
COR-EFPD-TOT	Total cumulative fission power and decay heat generated in core. (units = J)
COR-EMWR-TOT	Total cumulative oxidation heat generated in core. (units = J)
COR-EB4C-TOT	Total cumulative heat from B_4C reactions in core. (units = J)
COR-ECNV-TOT	Total cumulative energy transfer to CVH package. (units = J)
COR-EBND-TOT	Total cumulative energy transfer to HS package. (units = J)
COR-EFPD-RAT	Total fission power and decay heat generation rate in core. (units = W)
COR-EMWR-RAT	Total oxidation heat generation rate in core. (units = W)

COR-EB4C-RAT	Total heat generation rate from B_4C reactions in core. (units = W)
COR-ECNV-RAT	Total convective heat generation rate in core. (units = W)
COR-EBND-RAT	Total radiative heat transfer rate to boundary heat structures. (units = W)
COR-QCNV(NameCV)	Heat transfer rate (including radiation) to atmosphere in fluid volume NameCV. (units = W)
COR-RadHeatRateHS(IA)	Radiative heat transfer rate to radial boundary heat structure in level IA. (units = W)
COR-AxlHeatRateHS(IR)	Radiative heat transfer rate to axial boundary heat structure in ring IR. (units = W)
COR-HTCLH(IS)	Heat transfer coefficient from segment IS of lower head to water pool. (units = W/m^2K)
COR-HTCLH-AVE	Average heat transfer coefficient from lower head to water pool. (units = W/m^2K)
COR-QFLXLH(IS)	Heat flux from segment IS of lower head to water pool. (units = W/m^2)
COR-QFLXLH-AVE	Average heat flux from lower head to water pool. (units = W/m^2)
COR-QTOTLH(IS)	Cumulative heat transferred from segment IS of lower head to water pool. (units = J)
COR-QTOTLH-TOT	Total cumulative heat transferred from lower head to water pool. (units = J)

5.5 Core Structural Loading and Damage Variables

COR-SS-LOAD(IA,IR)	Load supported by supporting structure (SS) in cell (IA,IR). (units = kg)
COR-SS-STRESS(IA,IR)	Maximum component of stress in SS in core cell (IA,IR). (units = Pa)
COR-SS-DAMAGE(IA,IR)	Damage fraction (fraction of damage that will produce failure) for SS in core cell (IA,IR). (units = none)
COR-SS-TLEFT(IA,IR)	Remaining life for SS in core cell (IA,IR) (for unchanged temperature and load). (units = s)
COR-ROD-DAM(IA,IR)	Damage fraction for fuel rod in core cell (IA,IR). The fraction is based on life-time-remaining-at-temperature table input on COR_ROD record. If no table record is present on COR_ROD, the fraction is zero. (units = none)
COR-DAM-FLAG	Logical Fuel failure status. Returns .TRUE. if fuel failure has occurred in at least one core cell, as indicated by COR-ROD-DAM(IA,IR) exceeding 1.0. The CF variable requires life-time-remaining-at-temperature table input on COR_ROD record. If no table record is present on COR_ROD, COR-DAM-FLAG returns .FALSE. (units = none)

5.6 Core Oxidation and B_4C Reaction Mass Variables

COR-H2massProd(MAT)	Total cumulative hydrogen production by oxidation of material MAT in core. The choices allowed for MAT are SS, ZR, B4C or TOT for all oxidation processes. (units = kg)
COR-DMCO-TOT	Total cumulative carbon monoxide production in core. (units = kg)

COR-DMCO2-TOT	Total cumulative carbon dioxide production in core (available only for MCRP = B4C). (units = kg)
COR-DMCH4-TOT	Total cumulative methane production in core (available only for MCRP = B4C). (units = kg)
COR-ZROX-TLEFT(IA,IR)	Time left in lifetime for clad component in cell (IA,IR). (units = sec)
COR-ZROX-LIFE(IA,IR)	Accumulated damage for clad component in cell (IA, IR). (units = sec)
COR-EM(IA,IR,SURF)	Local emissivity of surface SURF in cell (IA,IR). (units = dimensionless)
COR-HTC(IA,IR,COMP,CS)	Local heat transfer coefficient to convective surface CS (CS=P for pool, CS=A for atmosphere, CS=S for Pool surface) for component COMP in cell (IA,IR). (units = J/m-K)
COR-OXTH(IA,IR,COMP,IM)	Local oxide thickness for component COMP in cell n. (units = m)
COR-H2(IA,IR,COMP)	Local hydrogen generation from component COMP in cell n (intact and conglomerate) (units = kg)
COR-H2C(IA,IR,COMP)	Local hydrogen generation from conglomerate on component COMP in cell n (units = kg)
COR-HTR-Q	
COR-CELLFCELR(IA,IR)	Local radial radiative exchange factor in cell n. (units = dimensionless)
COR-CELLFCELA(IA,IR)	Local axial radiative exchange factor in cell n. (units = dimensionless)
COR-TAVG	

5.7 Core Quench Variables

In the following three variables, a character string is required to define the component COMP. Acceptable strings are FU, CL, CN, CB, SS, NS, PB, PD, MP1, MP2, MB1, MB2, SH (for PWR), and FM (for PWR). (Interpretation is given by the table in Section 1.1.) As indicated, most are implemented only as control function arguments but can be made available for plotting by use of EXEC_PLOT input records. The variables for the cladding component, COR-ZQ(IA,IR,CL), COR-TQ(IA,IR,CL), and COR-TUQ(IA,IR,CL), are implemented as plot variables.

COR-ZQ(IA,IR,COMP)	Position of the quench front on the surface of component COMP in cell (IA,IR). It is returned as the elevation of the bottom of the cell if the entire surface is unquenched and as the elevation of the top of the cell if it is quenched. This is also available as a plot variable, for COMP =CL only. (units = m)
COR-TQ(IA,IR,COMP)	Temperature of the quenched portion of the surface of component COMP in cell (IA,IR). If there is no quench front in the core cell, it is returned as the average temperature. This is also available as a plot variable, for COMP =CL only. (units = K)
COR-TUQ(IA,IR,COMP)	Temperature of the unquenched portion of the surface of component COMP in cell (IA,IR). If there is no quench front in the core cell, it is returned as the average temperature. This is also available as a plot variable, for COMP =CL only. (units = K)

5.8 Core Penetration, Lower Head, Breach and Melt Ejection Variables

COR-TPN(n)	Temperature of penetration n. (units = K)

In the following three variables, (IN,IS) refers to node IN of segment IS, where IN = 01 is on the outside of the lower head. (Although nodal information on COR_LHN record is entered from inside to outside, the nodes are actually numbered in the opposite direction.)

COR-TLH(IN,IS)	Temperature of vessel lower head node. (units = K)

COR-VSTRAIN(IN,IS)	Plastic strain in vessel lower head node. (units = dimensionless)
COR-VSTRESS(IN,IS)	Stress in vessel lower head node. (units = Pa)
COR-ABRCH	Total flow area of vessel breach. (units = m^2)
COR-MEJEC-TOT	Total debris mass ejected through vessel breach. (units = kg)

5.9 Convecting Molten Pool Variables

In the following variables acceptable strings for material MAT are:

OXI - oxidic molten pool
MET - metallic molten pool

Acceptable strings for KEY are CH – channel, BY – bypass.

COR-T-LP(MAT,key)	Temperature of lower plenum convecting oxidic/metallic molten pool in channel or bypass. (units = K)
COR-T-UP(MAT,key)	Temperature of upper core convecting oxidic/metallic molten pool in channel or bypass. (units = K)
COR-M-LP(MAT,key)	Total mass of lower plenum convecting oxidic/metallic molten pool in channel or bypass. (units = kg)
COR-M-UP(MAT,key)	Total mass of upper core convecting oxidic/metallic molten pool in channel or bypass. (units = kg)
COR-V-UP(MAT,key)	Volume of upper core convecting oxidic/metallic molten pool in channel or bypass. (units = m)
COR-V-LP(MAT,key)	Volume of lower plenum convecting oxidic/metallic molten pool in channel or bypass. (units = m)

COR-RA-UP(MAT,key)	Rayleigh number of upper core convecting oxidic/metallic molten pool in channel or bypass. (units = m)
COR-RA-LP(MAT,key)	Rayleigh number of lower plenum convecting oxidic/metallic molten pool in channel or bypass. (units = m)
COR-MPR-UP(MAT,key,nCLS)	Mass of radionuclide class nCLS in upper oxide/metalic convecting molten pool above the core support plate and located in channel or bypass. (units = kg)
COR-MPR-LP(MAT,key,nCLS)	Mass of radionuclide class nCLS in lower oxide/metalic convecting molten pool above the core support plate and located in channel or bypass. (units = kg)

5.10 Core Accounting Variables

COR-DT	COR package timestep. (units = s)
COR-NCYCLE	Total number of COR package subcycles. (units = none)
COR-CPU	Total CPU usage by run portion of COR package. (units = s)
COR-MASSERR	Total mass error for COR package. (units = kg)
COR-ENERGYERR	Total energy error for COR package. (units = J)
COR-REL-ENGY-ERR	COR relative energy error, defined as COR-ENERGYERR/ COR-ENERGY-TOT. (units = dimensionless)
COR-REL-ENGY-ERM	COR relative energy error, modified to include enthalpy of mass transfers to BH and HS, defined as COR-ENERGYERR/(COR-ENERGY-TOT + debris to CAV).

	(units = dimensionless)
COR-MASS-DISCARD	Total accumulated mass of discarded components of negligible mass. (units = kg)
COR-ENGY-DISCARD	Total accumulated energy associated with discarded components of negligible mass. (units = J)

6 Example Input

6.1 Example MELGEN Input

The following are sample MELGEN input records for the COR package for a 3-ring by 12-level nodalization. It corresponds to the input example presented in previous versions of this guide, with the substitution of supporting and nonsupporting structures (SS and NS) for undifferentiated "other structure" (OS) and uses COR_Kcmp input records rather than COR_CCM records. Pending improvements to the core structures support model, a small amount of NS is required in level 1 to support the control rod guide tubes in levels 2 through 5. This example does not represent any real reactor system.

```
COR_INPUT   !  COR package start record
! GENERAL CORE INPUT
! COR_GP – Geometric Parameters
!         RFUEL     RCLAD    DRGAP    PITCH
COR_GP   .005207   .0061341 .0001143  .016
! COR_RT – Reactor Type
!        IRTYP  MCRP     MATHT
COR_RT   BWR    B4C
! COR_VP – Vessel Parameters
!      RCOR     RVESS    ILHTRN  DZRV    DZLH    ILHTYP      RVLH
COR_VP 2.430996 2.567837 RCOR    0.2408  .2254   HEMISPHERE  2.824621
! COR_AVP – Additional Vessel Parameters
!        HLST         HCSP
COR_AVP  4.0693E+01   4.1184E+01
! COR_RF – Radiative Exchange Factors
!         FCNCL    FSSCN    FCELR    FCELA    FLPUP
COR_RF    0.25     0.95     0.25     0.25     0.25
! COR_TP – TP Interface and
!          Fission Power and Gap Conductance Control Functions
!       NTPCOR  RNTPCOR ICFGAP ICFFIS     CFNAME
COR_TP  TPin101 TPin601 CF61   FIZPOWALL  CF51
! COR_CHT – Candling Heat Transfer Coefficients
!         HFRZZR   HFRZZX   HFRZUO   HFRZSS   HFRZSX   HFRZCP
COR_CHT   2000.0   2000.0   2000.0   2000.0   2000.0   2000.0
! COR_MS – Model Switches
COR_MS  ACTIVE ! IEUMOD
! COR_CMT – Candling Secondary Material Transport Parameters
!         MTUOZR MTZXZR MTSXSS MTCPSS FUOZR FZXZR FSXSS FCPSS
COR_CMT   1      2      2      2      0.20  1.00  1.00  0.00
! COR_LHF – Lower Head Failure Modeling Parameters
!       HDBPH/CF HDBLH/CF HMPOLH/CF HMPMLH/CF TPFAIL   CDISPN
COR_LHF 500.0    500.0    MODEL     MODEL     1273.15  1.0
```

```
! COR_ZP – Axial Level Parameters
COR_ZP   12 ! NAXL
! IA      Z       DZ       PORDP   IHSA      FZPOW
   1      0.0     1.2954   0.3     HS10001   0.0
   2      1.2954  0.9803   0.3     HS10002   0.0
   3      2.2757  0.9803   0.3     HS10003   0.0
   4      3.2560  0.9803   0.3     HS10004   0.0
   5      4.2363  0.9803   0.3     HS10005   0.0
   6      5.2166  0.2777   0.3     HS10006   0.0
   7      5.4943  0.635    0.3     HS10207   0.75
   8      6.1293  0.635    0.3     HS10208   0.9
   9      6.7643  0.635    0.3     HS10209   1.2
  10      7.3993  0.635    0.3     HS10210   1.2
  11      8.0343  0.635    0.3     HS10211   0.9
  12      8.6693  0.635    0.3     HS10212   0.75
! COR_RP – Radial Ring Parameters
COR_RP  3 ! NRAD
! IR  RINGR    ASCELA   IHSR     UNUSED  ICFCHN   ICFBYP   FRPOW
  1   1.35     5.729    HS10301      -        NO       NO        1.2
  2   1.91     5.729    HS10302      -      NO       NO        0.9
  3   2.80     5.729    HS10303      -      NO       NO       0.75
! SS AND NS OPTIONS
! TREAT ALL NS IN INPUT AS CONTROL BLADE
! COR_NS – Global Support Rule for NS
COR_NS 1
! N  IA IR  INSSUP METAL  TNSMAX  DRNSMN
  1 ALL ALL BLADE  STEEL  0.0     0.0001  ! default
! COR_SS – Global Loading and Failure Rule for SS
COR_SS   2 ! IA IR   ISSMOD
        1  ALL ALL  COLUMN 50. 0.14 0.13 ! XNUM RADO RADI
        2  ALL 6    PLATEB 0.08 0.3 0.15 ! THICK SPACE AKMB
! COR_RBV – Cell Reference and Fluid Boundary Volumes
COR_RBV 14 !  IA  IR    AXLR RADR ICVHC  ICVHB
        1     1    1    0    0    CV100  CV100
        2     1    2-3  1    1 ! REFERENCE TO CELL 101 FOR MISSING INPUT
        3     2    1    0    0    CV100  CV100
        4     3-5  1    2    1 ! REFERENCE TO CELL 102 FOR MISSING INPUT
        5     2    2-3  2    1 ! REFERENCE TO CELL 102 FOR MISSING INPUT
        6     3-5  2    2    2 ! REFERENCE TO CELL 202 FOR MISSING INPUT
        7     3-5  3    2    3 ! REFERENCE TO CELL 302 FOR MISSING INPUT
        8     6    1    0    0    CV100  CV100
        9     6    2-3  6    1 ! REFERENCE TO CELL 106 FOR MISSING INPUT
        10    7    1    0    0    CV101  CV102
        11    7    2-3  7    1 ! REFERENCE TO CELL 107 FOR MISSING INPUT
        12    8-12 1    7    1 ! REFERENCE TO CELL 107 FOR MISSING INPUT
        13    8-12 2    7    2 ! REFERENCE TO CELL 207 FOR MISSING INPUT
        14    8-12 3    7    3 ! REFERENCE TO CELL 307 FOR MISSING INPUT
! COR_CIT – Initial Component Temperatures
COR_CIT 4 ! n IA   IR TFU   TCL   TOS   TCN   TCB   TPD   TSS   TNS
            1 1-2  1  560.0 560.0 560.0 560.0 560.0 560.0 560.0 560.0
            2 6    1  560.0 560.0 560.0 560.0 560.0 560.0 560.0 560.0
            3 7    1  595.0 560.0 560.0 560.0 560.0 560.0 560.0 560.0
            4 8-12 1  595.0 560.0 560.0 560.0
! COR_EDR – Equivalent Diameter Record
COR_EDR 4 ! n IA IR   DHYCL DHYOS DHYDP DHYCNC DHYCNB DHYSS DHYNS
            1 1    1  1.0   1.0   0.005 1.0    1.0    1.0   1.0
            2 2    1  1.0   1.0   0.005 1.0    1.0    1.0   0.3
            3 6    1  1.0   1.0   0.005 1.0    1.0    0.15  1.0
            4 7    1  0.005 1.0   0.005 0.005  0.003  1.0   0.003
! COR_BFA – Cell Boundary and Flow Areas Record
COR_BFA 12 ! n IA   IR ASCELR   AFLOWC   AFLOWB
             1 1    1  10.99    5.729    0.0
             2 1    2  15.54
```

```
             3 1    3  21.98
             4 2    1  8.3177   5.5039   0.0
             5 2    2  11.763
             6 2    3  16.6354
             7 6    1  2.356    3.4585   0.0
             8 6    2  3.332
             9 6    3  4.712
            10 7    1  5.3879   2.65     0.8
            11 7    2  7.6196
            12 7    3  10.7758
! COR_SA – Surface Area Record
COR_SA 4 ! n IA IR  ASFU  ASCL  ASOS ASCN  ASSS  ASNS
           1 1   1  0.0   0.0   0.0  0.0   0.0   0.0
           2 2   1  0.0   0.0   0.0  0.0   106.1 0.0
           3 6   1  0.0   0.0   0.0  0.0   50.0  0.0
           4 7   1  328.0 399.0 0.0  86.7  0.0   127.0
! COR_KSS – Supporting Structure Component Masses
COR_KSS 2 ! n IA  IR  XMSSSS
            1 2    1  1750.0
            2 6    1  5000.0
! COR_KFU – Fuel Component Masses
COR_KFU 2  ! n    IA IR   XMFUUO
             1     1  1   0.0
             1     7  1   9360.0
! COR_KCL – Cladding Component Masses
COR_KCL 1 ! n IA IR
            1 7   1  1968.0
! COR_KNS – Nonsupporting Structure Component Masses
COR_KNS 1 ! n IA IR
            1 7   1  596.0 99.2
! COR_KCN – Canister (CN) Component Masses
COR_KCN 1 ! n IA IR
            1 7   1  730.0
! COR_KCB – Canister (CB) Component Masses
COR_KCB 1 ! n IA IR
            1 7   1  730.0
! COR_LH – Lower Head Structure Record
COR_LH 5
! COR_LHD – Lower Head Segments Record
COR_LHD   3   3          !     NLHT  NLHTA
!         IS  TLH    RADLH     NameCVH
          1   560.  1.350     CV205
          2   560.  1.910     CV205
          3   560.  2. 80     CV205
! COR_LHN – Lower Head Nodalization Record
! THE FOLLOWING RECORDS DEFINE A HEAD WITH 1 INCH OF STAINLESS
! STEEL INSIDE A CARBON STEEL SHELL, USING UNEQUALLY-SPACED NODES.
COR_LHN 4 ! n MATLH           DZHEAD    MATCYL          DZWALL
            1 STAINLESS-STEEL 0.0254    STAINLESS-STEEL 0.0254
            2 CARBON-STEEL    0.0754    CARBON-STEEL    0.0754
            3 CARBON-STEEL    0.1754    CARBON-STEEL    0.1754
            4 CARBON-STEEL    0.2254    CARBON-STEEL    0.2254
! COR_PEN – Lower Head Penetration Record
COR_PEN 3 ! n ref is  XMPN   TPN    ASPN  AXPN   AFLPN ICFLHF
            1  0  1   704.0  560.0  18.3  0.431  1.03  PenFail
            2  1  2
            3  1  3
```

The following are sample MELGEN input records for the DCH and RN packages to treat the additional reaction products generated when the advanced B_4C reaction model is selected by overriding the default (simple model) selection with sensitivity coefficient 1005. RN

classes 17–23 have been defined to treat the additional reaction products, and the DCH input assigns zero decay heat to those classes.

```
!*************************
!* DECAY HEAT DEFINITION *
!*************************
DCH_input
DCH_RCT BWR
! USE ALL DEFAULT RADIONUCLIDE CLASSES
! SELECT ALL DEFAULT CLASSES
DCH_CL 'XE' DEFAULT
DCH_CL 'CS' DEFAULT
DCH_CL 'BA' DEFAULT
DCH_CL 'I2' DEFAULT
DCH_CL 'TE' DEFAULT
DCH_CL 'RU' DEFAULT
DCH_CL 'FE' DEFAULT
DCH_CL 'CE' DEFAULT
DCH_CL 'LA' DEFAULT
DCH_CL 'UO2' DEFAULT
DCH_CL 'CD' DEFAULT
DCH_CL 'AG' DEFAULT
DCH_CL 'BO2' DEFAULT
DCH_CL 'H2O' DEFAULT
DCH_CL 'CON' DEFAULT
DCH_CL 'CSI' DEFAULT
! new elements input
DCH_EL CI  1.e-9    1
                    1 0.0   0.0
DCH_EL B2  1.e-9    1
                    1 0.0   0.0
DCH_EL B3  1.e-9    1
                    1 0.0   0.0
DCH_EL B4  1.e-9    1
                    1 0.0   0.0
DCH_EL B5  1.e-9    1
                    1 0.0   0.0
DCH_EL B6  1.e-9    1
                    1 0.0   0.0
DCH_EL B7  1.e-9    1
                    1 0.0   0.0
DCH_EL B8  1.e-9    1
                    1 0.0   0.0
! redefine CSI class
DCH_CL 'CSI'    USER 1
                     1 CI
! define new classes
DCH_CL 'H3B3O6' USER 1
                     1 B2
DCH_CL 'HBO2'   USER 1
                     1 B3
DCH_CL 'BH3'    USER 1
                     1 B4
DCH_CL 'B2H6'   USER 1
                     1 B5
DCH_CL 'BOH'    USER 1
                     1 B6
DCH_CL 'B(S)'   USER 1
                     1 B7
DCH_CL 'C(S)'   USER 1
                     1 B8
!*********************
```

```
!* RADIONUCLIDE INPUT *
!**********************
*
! ACTIVATE RN1 PACKAGE
RN1_input
! RN1_DIM Record – Dimension Record
RN1_DIM 5 1 23 !  NUMSEC NUMCMP NUMCLS
```

6.2 Example MELCOR Input

The following are sample MELCOR input records for the COR package.

```
! COR_DTC – COR Package Time Step Control Record
!       DTMPCR DTCMIN  NSUBMX
COR_DTC 30.0   0.005   32
! COR_EDV – Edit Flags Record
!       ITEMP IMASS IVOL IASUR IPMV
COR_EDV 1     1     1    0      0
! COR_TST – Physics On/Off Switches
!       IRAD ICND ICNV IOXD IDRP ITDZ IB4C IRDS IDEJ ISPR
COR_TST 0    0    0    0    0    0    1    0    1    0
```

7 COR Package Output

The COR package output provides a description of the current state of the reactor core, including any debris. Many two-dimensional arrays dimensioned (NAXL, NRAD) are printed in a matrix format from the top downward to aid in visualizing the physical processes modeled in the core.

For a MELGEN run and on the first cycle of a MELCOR restart, all the available information from the COR package is printed, including much of the time-independent database. By default, only temperatures and masses, along with energy and mass balances and some statistical information, are written for succeeding edit cycles. However, the user may also specify the editing of various geometric variables, such as surface areas or cell flow areas, or may eliminate nearly all of the COR package edit information through the use of the edit flags specified on the COR_EDV MELCOR input record.

Special messages are written to the message file at the time of failure for each supporting structure and each lower head penetration, at the beginning and end of the falling debris quench calculation in each ring, and at the beginning of debris ejection to the cavity.

Complete output for a sample problem can be found in the output listing for the demonstration problem distributed with MELCOR.

8 Diagnostics and Error Messages

Diagnostics and error messages generated during MELGEN are concerned with input processing and are generally self-explanatory. If these messages seem obscure or are otherwise unhelpful in isolating an input problem, please contact the MELCOR code development group.

A variety of messages may be generated during the execution phase of MELCOR. Some of these are merely informative and should not be of much concern unless they occur very frequently during the calculation. Others indicate very serious problems in the calculation and should be referred to the MELCOR code-development group. If a calculation aborts (gracefully or not), please save all input, output, plot, and restart files for use by the code developers in resolving the error.

Nonfatal Diagnostics:

(1) WARNING FROM COR PACKAGE
ENERGY ERROR IS BECOMING LARGE
The relative energy error has exceeded a preset limit (currently 5.E-5). The magnitude of this error is indicative of its importance in the calculation and should normally be within a factor of 100 of machine roundoff. Any error over 5.E-4 should be reported to the MELCOR code development group.

(2) WARNING FROM COR PACKAGE
MASS ERROR IS BECOMING LARGE
The relative mass error has exceeded a preset limit (currently 5.E-5). The magnitude of this error is indicative of its importance in the calculation and should normally be within a factor of 100 of machine roundoff. Any error over 5.E-4 should be reported to the MELCOR code development group.

(3) WARNING FROM COR PACKAGE
VOLUME ERROR IS BECOMING LARGE
The relative volume error has exceeded a preset limit (currently 5.E-5). The magnitude of this error is indicative of its importance in the calculation and should normally be within a factor of 100 of machine roundoff. Any error over 5.E-4 should be reported to the MELCOR code development group.

(4) WARNING FROM COR PACKAGE
INTERNAL ENERGIES DID NOT CONVERGE IN CELL (IA,IR), COMPONENT COMP
DECOR=x.xxxxEyy,SUMMCP=x.xxxxEyy,TNEW=xxxx.xx,ENEW=x.xxxEyy
Difficulty was encountered in equilibrating the internal energies of the materials within a cell component. The COR package continues the calculation by repeating the subcycle with a smaller timestep. Frequent occurrences of this message should be reported to the MELCOR code development group.

(5) CORRAD – Implied View Factor > 1
s1 IN CELL (IA1,IR1) <-> s2 IN CELL (IA2,IR2)
As a result of inappropriate input or change in geometry, some view factor implied by reciprocity (AiFij = AjFji) exceeds unity.
Here, s may be any of the surfaces, including PELLET, CLAD, CAN, CAN-B, OTH.STR, PRT.DBR, CAN.OUT, CAN-B.O, RAD.BND, or AXL.BND, where, AXL.BND, and RAD.BND refer to an unnamed component in one cell that provides an axial or radial boundary for a named component in another. Axial level

NAXL+1 and radial ring NRAD+1 refer to the heat structures providing the radiative boundary condition specified by input record COR_RP or COR_ZP, respectively.

(6) VIEW FACTOR ERROR IN CORVF
SUM OF VIEW FACTORS EXCEEDS UNITY FOR SURFACE s IN CORE CELL (IA,IR)
The sum of view factors input or implied for a single surface may exceed unity. This is not a particularly serious problem because only pairs of surfaces are considered. (MELCOR does not use a net enclosure model.)

In neither case 5 nor case 6 will the message be repeated more than 50 times; the last will include a final line—

FURTHER MESSAGES SUPPRESSED

If such warnings are seen in MELGEN diagnostics (under conditions of intact geometry), user input should be reexamined and, in most cases, modified.

(7) VOLUME CONSISTENCY WARNING IN CELL (IA,IR) at location
Channel volume error = x.xxxEeee (m3)
Bypass volume error = x.xxxEeee (m3)
Net volume error = x.xxxEeee (relative)
Here "at location" indicates the point in the advancement through the timestep where the problem was detected. This message normally represents an incursion of channel components into the bypass region of a BWR, or vice versa, by more than the limit set by Sensitivity coefficient C1504(2). It is not a significant cause of concern unless the magnitude is large.

(8) VOLUME CONSISTENCY ERROR IN CELL (IA,IR) at location
Net volume error = x.xxxEeee (relative)
Here "at location" indicates the point in the advancement through the timestep where the problem was detected. This message is a result of an inconsistency in the internal representation of volumes in the COR package by more than the limit set by Sensitivity coefficient C1504(1). It is indicative of a more severe error than the preceding message.

In neither case 7 nor case 8 will the message be repeated more than 100 times; the last will include a final line—

FURTHER MESSAGES SUPPRESSED

<u>Fatal Error Messages:</u>

(1) TOO MANY SUBCYCLES OR TOO SMALL A TIME STEP IN COR PACKAGE or SUBCYCLE TIME STEP IN COR PACKAGE IS TOO SMALL
Too many COR package subcycles were generated during a single system cycle, or the subcycle timestep was reduced to below the user-defined minimum. User

input variables on record COR_DTC can be adjusted, but if the COR package then consumes an excessive amount of CPU time or if the error message reoccurs, it should be reported to the MELCOR code development group.

(2) ERROR IN SUBROUTINE CORUVN
TOTAL MASS IS ZERO IN COMPONENT COMP OF CELL (IA,IR)
ATTEMPT TO ADD ENERGY = x.xxxxEyy
An error has been detected during an attempt to transfer energy to or from a cell component that has zero mass. This error should be reported to the MELCOR code development group.

(3) CONTROL FUNCTION ERROR IN SUBROUTINE CORabc
An error flag was set by the CF package during a call from subroutine CORabc. The user input for the control function should be checked. If the error cannot be resolved, it should be reported to the MELCOR code development group.

(4) TABULAR FUNCTION ERROR IN SUBROUTINE CORabc
An error flag was set by the Tabular Function (TF) package during a call from subroutine CORabc. The user input for the tabular function should be checked. If the error cannot be resolved, it should be reported to the MELCOR code development group.

(5) VOLUME CONSISTENCY ERROR IN CELL (IA,IR) at location
Net volume error = x.xxxEeee (relative)
MUST INCREASE VOLT=SC1504(1) TO CONTINUE
Here "at location" indicates the point in the advancement through the timestep where the problem was detected. This message is a result of an inconsistency in the internal representation of volumes in the COR package by more than 100 times the limit set by sensitivity coefficient C1504(1). It is indicative of a severe error and should be reported to the MELCOR code development group.

(6) ERROR IN SUBROUTINE CORabc
An error was detected in subroutine CORabc. Any errors of this form should be reported to the MELCOR code development group.

Appendix 1

Melcor 2.0	PTFRead	Comments
Cor-CellMass(IA,IR,mat) IA – the axial level number; IR – the radial ring number; mat – the material name: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-MMAT.n n = 100*IA+IR; MAT - the material name: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 – zirconium dioxide.	Total mass of material MAT in cell (IA,IR) (n).
Cor-TotMass(mat) mat – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-MMAT MAT – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Total mass of material MAT in core.
Cor-DC(IA,IR,COMP,mat) IA – the axial level number; IR – the radial ring number; COMP – the component name. The choices allowed for COMP are: FU, CL, CN, CB, SS, NS, SH, FM, PB, PD, MP1, MP2, MB1, MB2. mat – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-DC.m.k.n n = 100*IA+IR; k – the component name. The choices allowed for k are: FU, CL, CN, CB, SS, NS, SH, FM, PB, PD, MP1, MP2, MB1, MB2. m – the material name. The choices allowed for m are: MCRP - control poison , MINC - Inconel, MSS - steel, MSSOX - steel oxide, MUO2 – uranium dioxide, MZR - zircaloy, MZRO2 - zirconium dioxide.	Conglomerate debris mass of material MAT (m) on component COMP (k) in cell (IA,IR) (n).
Cor-PB(IA,IR,mat) IA – the axial level number; IR – the radial ring number; mat – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-PB.m.n n = 100*IA+IR; m – the material name. The choices allowed for m are: MCRP - control poison , MINC - Inconel, MSS - steel, MSSOX - steel oxide, MUO2 – uranium dioxide, MZR - zircaloy, MZRO2 - zirconium dioxide.	Particulate debris mass of material MAT (m) in bypass of cell (IA,IR) (n).

Melcor 2.0	PTFRead	Comments
Cor-PD(IA,IR,mat) IA – the axial level number; IR – the radial ring number; mat – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-PD.m.n n = 100*IA+IR; m – the material name. The choices allowed for m are: MCRP - control poison , MINC - Inconel, MSS - steel, MSSOX - steel oxide, MUO2 – uranium dioxide, MZR - zircaloy, MZRO2 - zirconium dioxide.	Particulate debris mass of material MAT (m) in cell (IA,IR) (n).
Cor-M(IA,IR,COMP,mat) IA – the axial level number; IR – the radial ring number; COMP – the component name. The choices allowed for COMP are: FU, CL, CN, CB, SS, NS, SH, FM, PB, PD, MP1, MP2, MB1, MB2. mat – the material name. The choices allowed for MAT are: CRP - control poison , INC - Inconel, SS - steel, SSOX - steel oxide, UO2 – uranium dioxide, ZR - zircaloy, ZRO2 - zirconium dioxide.	Cor-M. m.k.n n = 100*IA+IR; k – the component name. The choices allowed for k are: FU, CL, CN, CB, SS, NS, SH, FM, PB, PD, MP1, MP2, MB1, MB2. m – the material name. The choices allowed for m are: MCRP - control poison , MINC - Inconel, MSS - steel, MSSOX - steel oxide, MUO2 – uranium dioxide, MZR - zircaloy, MZRO2 - zirconium dioxide.	Total mass (intact plus conglomerate) of material MAT (m) on component COMP (k) in cell (IA,IR) (n).
Cor-CellVolFrac (IA,IR,COMP) IA – the axial level number; IR – the radial ring number; COMP – the component name. The choices allowed for COMP are: FU, CL, CN, SS, NS, SH, FM, MP1, MB1, MP2, MB2, FL, FLB, FLC, RK, POR, PORB.	Cor-VOLF-COMP.n n = 100*IA+IR; COMP – the component name. The acceptable argument names are: Cor-VOLF-FU.n Cor-VOLF- CL.n Cor-VOLF-CN.n Cor-VOLF-SS.n Cor-VOLF- NS.n Cor-VOLF-SH.n Cor-VOLF-FM.n Cor-VOLF-MP1.n Cor-VOLF-MB1.n Cor-VOLF-MP2.n Cor-VOLF-MB2.n Cor-VOLF-FL.n Cor-VOLF-FLB.n Cor-VOLF-FLC.n Cor-VOLF-RK.n Cor-VOLF-POR.n Cor-VOLF-PORB.n	Volume Fraction of component COMP in cell (IA,IR) (n).

Melcor 2.0	PTFRead	Comments
Cor-CellTemp (IA,IR,COMP) IA – the axial level number; IR – the radial ring number; COMP – the component name. The choices allowed for COMP are: FU, CL, CN, CB, SS, NS, PB, PD, SH, FM, MP1, MB1, MP2, MB2, SVC, SVB, RK.	Cor-TCOMP.n n = 100*IA+IR; COMP – the component name. The acceptable argument names are: Cor-TFU.n Cor-TCL.n Cor-TCN.n Cor-TCB.n Cor-TSS.n Cor-TNS.n Cor-TPB.n Cor-TPD.n Cor-TSH.n Cor-TFM.n Cor-TMP1.n Cor-TMB1.n Cor-TMP2.n Cor-TMB2.n Cor-TSVC.n Cor-TSVB.n Cor-TRK.n	Temperature of component COMP in cell (IA,IR) (n).
COR-ENERGY-TOT	COR-ENERGY-TOT	Total internal energy in COR package.
COR-EFPD-TOT	COR-EFPD-TOT	Total cumulative fission power and decay heat generated in core.
COR-EMWR-TOT	COR-EMWR-TOT	Total cumulative oxidation heat generated in core.
COR-EB4C-TOT	COR-EB4C-TOT	Total cumulative heat from B_4C reactions in core.
COR-ECNV-TOT	COR-ECNV-TOT	Total cumulative energy transfer to CVH package.
COR-EBND-TOT	COR-EBND-TOT	Total cumulative energy transfer to HS package.
COR-EFPD-RAT	COR-EFPD-RAT	Total fission power and decay heat generation rate in core.
COR-EMWR-RAT	COR-EMWR-RAT	Total oxidation heat generation rate in core.
COR-EB4C-RAT	COR-EB4C-RAT	Total heat generation rate from B_4C reactions in core.
COR-ECNV-RAT	COR-ECNV-RAT	Total convective heat generation rate in core.
COR-EBND-RAT	COR-EBND-RAT	Total radiative heat transfer rate to boundary heat structures.
COR-QCNV(NameCV) NameCV – the name of control volume.	COR-QCNV.m m – the index of control volume.	Heat transfer rate (including radiation) to atmosphere in fluid volume NameCV (m).
COR-HTCLH(IS) IS – the segment number.	COR-HTCLH.IS IS – the segment number.	Heat transfer coefficient from segment IS of lower head to water pool.

Melcor 2.0	PTFRead	Comments
COR-HTCLH-AVE	COR-HTCLH-AVE	Average heat transfer coefficient from lower head to water pool.
Cor-QFLXLH(IS) IS – segment number.	COR-QFLXLH.IS IS – segment number	Heat flux from segment IS of lower head to water pool.
COR-QFLXLH-AVE	COR-QFLXLH-AVE	Average heat flux from lower head to water pool.
Cor-QTOTLH(IS) IS – segment number.	COR-QTOTLH.IS IS – segment number.	Cumulative heat transferred from segment IS of lower head to water pool.
COR-QTOTLH-TOT	COR-QTOTLH-TOT	Total cumulative heat transferred from lower head to water pool.
Cor-SS-STRESS(IA,IR) IA – the axial level number; IR – the radial ring number.	COR-SS-STRESS.ijj ijj = 100*IR+IA;	Maximum component of stress in SS in core cell (IA,IR) (ijj).
Cor-SS-DAMAGE(IA,IR) IA – the axial level number; IR – the radial ring number.	COR-SS-DAMAGE.ijj ijj = 100*IR+IA;	Damage fraction (fraction of damage that will produce failure) for SS in core cell (IA,IR) (ijj).
Cor-SS-TLEFT(IA,IR) IA – the axial level number; IR – the radial ring number.	COR-SS-TLEFT.ijj ijj = 100*IR+IA;	Remaining life for SS in core cell (IA,IR) (ijj) (for unchanged temperature and load).
COR-H2massProd(MAT) MAT – the material name. The choices allowed for MAT are: SS – stainless steel ZR - zircaloy B4C – boron carbide TOT – all oxidation processes	COR-DMH2-MAT MAT – the material name. The acceptable names for arguments are: COR-DMH2-SS; COR-DMH2-ZIRC; COR-DMH2-B4C; COR-DMH2-TOT.	Total cumulative hydrogen production by oxidation of material MAT in core.
COR-DMCO-TOT	COR-DMCO-TOT	Total cumulative carbon monoxide production in core.
COR-DMCO2-TOT	COR-DMCO2-TOT	Total cumulative carbon dioxide production in core (available only for MCRP = 'B4C'))..
COR-DMCH4-TOT	COR-DMCH4-TOT	Total cumulative methane production in core (available only for MCRP = 'B4C').
COR-TPN(n) n – the penetration number.	COR-TPN.n n – the penetration number.	Temperature of penetration n
COR-TLH(IN,IS) IN – node index; IS – segment index.	COR-TLH.ijj ijj = 100*IS+IN;	Temperature of vessel lower head node.
COR-VSTRAIN(IN,IS) IN – node index; IS – segment index.	COR-VSTRAIN.ijj ijj = 100*IS+IN;	Plastic strain in vessel lower head node.
COR-VSTRESS(IN,IS) IN – node index; IS – segment index.	COR-VSTRESS.ijj ijj = 100*IS+IN;	Stress in vessel lower head node.
COR-MEJEC-TOT	COR-MEJEC-TOT	Total debris mass ejected through vessel breach.
COR-DT	COR-DT	COR package time step.
COR-NCYCLE	COR-NCYCLE	Total number of COR package subcycles.
COR-CPU	COR-CPU	Total CPU usage by run portion of COR package.

Melcor 2.0	PTFRead	Comments
COR-MASSERR	COR-MASSERR	Total mass error for COR package.
COR-ENERGYERR	COR-ENERGYERR	Total energy error for COR package.
COR-REL-ENGY-ERR	COR-REL-ENGY-ERR	COR relative energy error, defined as COR-ENERGYERR/ COR-ENERGY-TOT
COR-REL-ENGY-ERM	COR-REL-ENGY-ERM	COR relative energy error, modified to include enthalpy of mass transfers to BH and HS, defined as COR- ENERGYERR / (COR-ENERGY-TOT + debris to CAV)

Control Volume Hydrodynamics (CVH) Package Users' Guide

The Control Volume Hydrodynamics (CVH) and Flow Path (FL) packages are responsible for modeling the thermal-hydraulic behavior of liquid water, water vapor, and gases in MELCOR. Modeling is based on a control volume/flow path formulation and is described in detail in the Thermal Hydraulic (CVH and FL) Package's Reference Manual.

This Users' Guide describes input to the CVH package, which is concerned with the geometry and contents of control volumes. CVH input describes the initial state of each volume; optional input allows the definition of sources and sinks of mass and/or energy, in terms of data from the Control Function (CF) and Tabular Function (TF) packages. Additional sources and sinks may be imposed by other packages during a MELCOR calculation; these are treated automatically and do not require input to CVH. Similarly, changes in control volume geometry resulting from relocation of nonhydrodynamic materials are communicated to CVH during MELCOR execution; no CVH input is required.

Connections between control volumes, through which the control volume contents may flow, are defined by input to the FL package, which is described in the FL Package Users' Guide.

Contents

LIST OF FIGURES **4**

LIST OF TABLES **4**

1 INTRODUCTION **5**

1.1 Control Volume Geometry 5
1.2 Control Volume Contents 6
1.3 Pool/Atmosphere Mass and Energy Transfer 8
1.3.1 Solution Algorithm and Numerical Controls 8
1.3.2 Conduction/Convection-Driven Transfer 8
1.3.3 Flashing Mode Mass and Energy Transfer 9
1.4 Mass and Energy Sources and Sinks 9
1.4.1 Flashing of Superheated Water Sources 11
1.5 Boundary Volumes 11
1.5.1 Volume Velocities for Boundary Volumes 12

2 USER INPUT **13**

2.1 MELGEN User Input 13
CV_DFT – CVH Package Default Scheme Record 13
CV_TENDINI – Time to end initialization 14
CVH_ATMCS – Atmosphere Sound Speed Model Selection 14
CVH_CSTBL – Print option to output a special table for the sound speed studies 14
CV_ID – Control Volume Name and Sequence Number 15
CV_DELETE – Ignore all records associated with CV_ID 15
CV_TYP – Control Volume Type 16
CV_THR – Switches for Thermodynamic Input 16
CV_VEL – Control Volume Initial Velocities 17
CV_ARE – Control Volume Area for Velocity Calculation 18
2.1.1 Thermodynamic input 19
CV_PAS – Pool and Atmosphere Switches 20
Mass Fraction Input (ITYPTH = 2) 21
CV_PTD – Control Volume Pressure or Temperature Data 21
CV_PAD – Pool Additional Data 22
CV_AAD – Atmosphere Additional Data 22
CV_BND – Boundary between the Pool and the Atmosphere 23
Separate Pool and Atmosphere Input (ITYPTH = 3) 23
CV_THERM – Alternative Input for Thermodynamic Data 23

CV_PTD – Control Volume Pressure or Temperature Data 27
CV_PAD – Pool Additional Data..... 28
CV_VOID – Void Fraction in the Pool..... 28
CV_AAD – Atmosphere Additional Data 29
CV_NCG – NCG Data Table 30
CV_FOG – Fog Data 32
CV_BND – Boundary between the Pool and the Atmosphere..... 32
CV_VAT – Altitude/Volume Table 36
CV_SOU – External Mass and Energy Sources..... 37
CV_DIF – Gas Diffusion Model..... 43

2.2 MELCOR User Input 43

CVH_TRACE – Calculational Trace Activation..... 43
ALLOWCOLDATM – Allow dry atmosphere temperature below 273.15..... 44

3 SENSITIVITY COEFFICIENTS.....44

3.1 CVH Sensitivity Coefficients..... 44

CVH_SC – CVH Sensitivity Coefficients Input Record 44
4400 – Timestep Control 45
4401 – Velocity Convergence Criteria 47
4402 – Minimum Velocity to be Considered for Choking Calculation 48
4404 – Friction Factor Parameters 48
4405 – SPARC Bubble Physics Parameters 49
4406 – Maximum Allowed Fog Density 50
4407 – Pool/Atmosphere Heat/Mass Transfer Parameters 50
4408 – Pressure Iteration Parameters..... 52
4409 – Limits and Tolerances for Time-Specified Volumes..... 53
4410 – Vapor Velocity Enhancement during Direct Containment Heating..... 53
4411 – Limits and Tolerances for Iterations in the CVT Package 54
4412 – Limits and Tolerances for Iterations in the CVH Package..... 55
4413 – Flow Blockage Friction Parameters 55
4414 – Minimum Hydrodynamic Volume Fraction 56
4415 – Criteria for Solving the Flow Equations in Sparse Form 56
4500 – Parameters in the Flashing Model for Sources and Flows..... 59

4 PLOT VARIABLES.....60

5 CONTROL FUNCTION ARGUMENTS 63

6 EXAMPLE INPUT.....66

7 CVH PACKAGE OUTPUT.....69

8 DIAGNOSTICS AND ERROR MESSAGES.....70

8.1 Input and Execution Messages 70
8.2 CVHTRACE Option 71
8.2.1 Main Advancement 73
8.2.2 Flow Solution 75

APPENDIX 1 77

List of Figures

Figure 1.1 Spatial Geometry (left) and Volume/Altitude Table (right). 5
Figure 1.2 Total, Hydrodynamic, and Virtual Volume. 6
Figure 1.3 Location of Material in Control Volume. 7

List of Tables

Table 2.1 Input Fields for Each Source Type. 42

1 Introduction

All hydrodynamic material, together with its associated energy, in a MELCOR calculation resides in control volumes. Hydrodynamic material includes the coolant (water), vapor and fog, and noncondensible gases; it does not include core structures or core debris, other heat structures, fission products, aerosols, water films on heat structures, or ice in ice condensers.

1.1 Control Volume Geometry

The spatial geometry of a control volume is defined by a volume/altitude table. Each point in the table gives an altitude and the cumulative volume available to hydrodynamic materials in the Control Volume Hydrodynamics (CVH) package below that altitude in that control volume. The volume at the lowest altitude (the bottom of the control volume) must be zero; the difference in altitude between the last point and the first one defines the height of the control volume, and the last volume entry defines its total volume. Volume is assumed to be a linear function of altitude between the table entries (see Figure 1.1). There may be an arbitrary number of entries in the table.

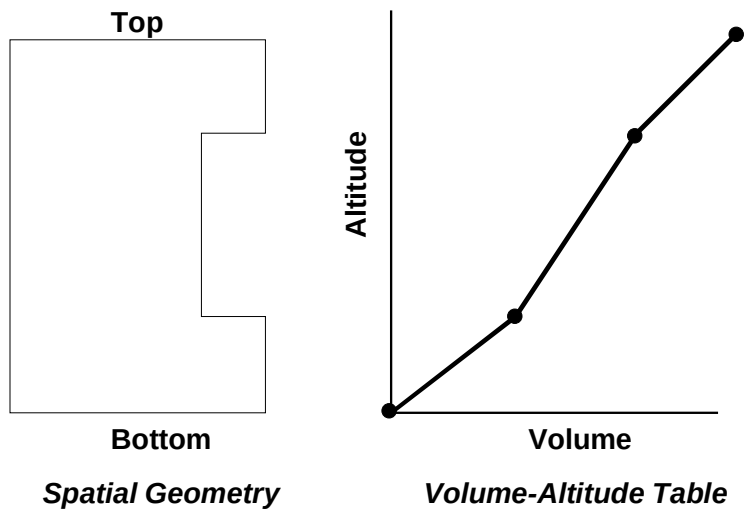


Figure 1.1 Spatial Geometry (left) and Volume/Altitude Table (right).

All altitudes input to MELCOR must be relative to a single, common reference point to allow determination of the difference in elevations of control volumes and of the location of connections to flow paths. The same reference point must be used in the CVH, Flow Path (FL), Core (COR), and Heat Structure (HS) packages. Any consistent reference point may be used; consequently, both positive and negative values of altitude are allowed.

CVH input defines the volume initially available to hydrodynamic materials; it should not include either the volume occupied by intact core components or that occupied by ice in an ice condenser. Such volume is referred to as virtual volume and represents space occupied by material in another MELCOR package (and, therefore, denied to hydrodynamic materials) that is subject to relocation. For example, when the core melts down and

relocates to the lower plenum, some or all of the volume that it initially occupied will be freed, but the resulting debris will occupy space in the lower plenum. In addition to the core, core debris, and ice, water films on surfaces can also occupy virtual volume.

Initial values of virtual volume are defined by the package (COR or HS) that models the associated material. An initial virtual volume/altitude table is constructed using the set of altitudes defined by CVH input and added to the table defined by CVH input to define a total volume/altitude table that does not change during a MELCOR calculation. However, the virtual volume/altitude table changes as nonhydrodynamic material is relocated, and the volume available for the hydrodynamic material, defined as the difference between the total and virtual volumes, changes as a function of time. This is illustrated in Figure 1.2, where core debris has been moved into the volume of Figure 1.1. Note that the tables are always based on the altitudes defined in CVH.

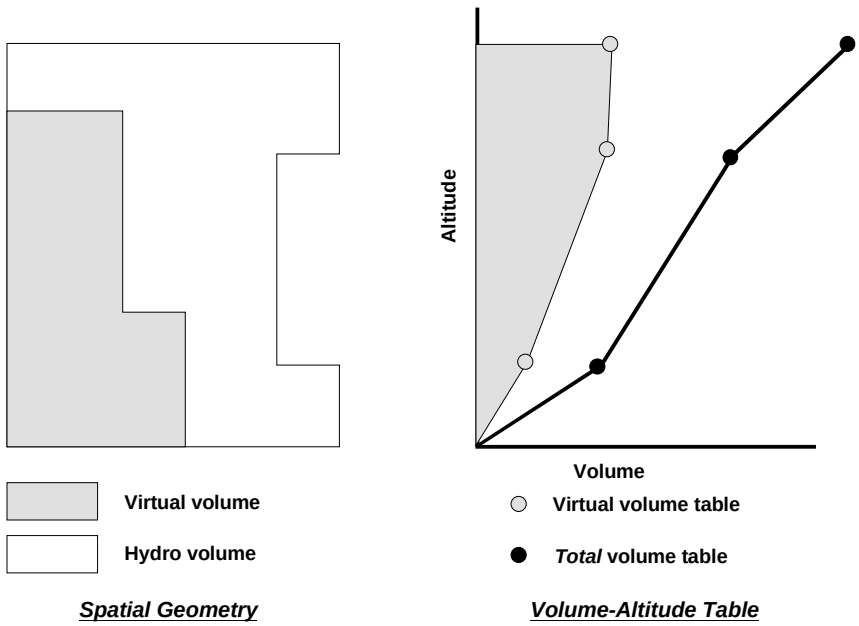


Figure 1.2 Total, Hydrodynamic, and Virtual Volume.

1.2 Control Volume Contents

The contents of each control volume are divided between a pool and an atmosphere, as shown in Figure 1.3. The pool is depicted as filling the lower portion of the control volume and the atmosphere filling the remainder. As discussed in the Thermal Hydraulic (CVH and FL) Packages Reference Manual, this figure is not interpreted so narrowly that it invalidates use of MELCOR hydrodynamics in other situations.

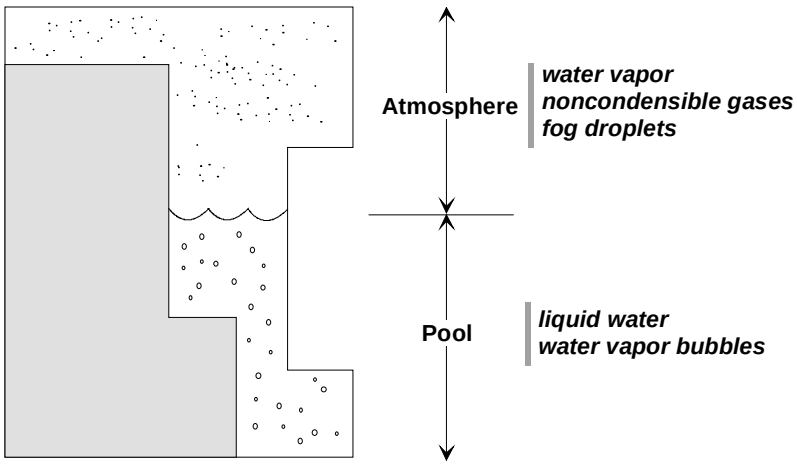


Figure 1.3 Location of Material in Control Volume.

The pool can be single-phase liquid water or two-phase water. The atmosphere contains water vapor and/or noncondensible gases. The atmosphere may also have suspended water droplets (referred to as fog). Although the interactions between rising gases and the pool may be modeled (see SPARC input in the FL Users' Guide), current modeling does not permit noncondensible gases to reside in the pool. The total volume is divided among the pool, the gaseous atmosphere, and the fog.

Materials are numbered within MELCOR. Materials 1, 2, and 3 are always the pool, the fog, and water vapor in the atmosphere, respectively. Materials with numbers greater than three (if included) are noncondensible gases and must be defined by input to the NonCondensible Gas (NCG) package.

The user can define the thermodynamic state of the material in the control volume in one of two ways:

(1) Specify the control volume pressure and the partial pressure of the water vapor in the atmosphere, the pool and atmosphere temperatures, water distribution in terms of mass fractions, and the mass fractions of noncondensible gases.

(2) Specify sufficient properties of the pool and the atmosphere to uniquely define—but not over-define—their thermodynamic states.

The first option is retained for consistency with older versions of MELCOR and is not recommended for new calculations as it provides no unique capabilities and its use will often require extensive hand calculations in reference to steam tables and/or trial-and-error iteration in MELGEN. In all cases, the input is checked for completeness and consistency. A number of details are available to reduce the amount of information that must be supplied to define, for example, a saturated volume. The second option is the default and will be used in most cases.

1.3 Pool/Atmosphere Mass and Energy Transfer

CVH input allows two choices for thermodynamic modeling in each control volume, referred to as equilibrium and nonequilibrium; the user should be aware that the latter is not the same as the "nonequilibrium" models in codes such as TRAC and RELAP5. Details may be found in the Control Volume Thermodynamics (CVT) Package Reference Manual. We recommend that the nonequilibrium option be used in general; the equilibrium option is retained primarily for historical reasons but may be appropriate in special cases.

When nonequilibrium thermodynamics option is used in a control volume, mass and energy transfers between the pool and the atmosphere are determined by the thermomechanical state of the materials in the control volume. In particular, these transfers depend on the temperatures of the pool and the atmosphere, the materials in the atmosphere, the velocity of the materials, etc. The CVH package calculates this energy and mass transfer between the pool and the atmosphere explicitly. The transfer is decomposed into a conduction/convection mode and a boiling/flashing mode.

When equilibrium thermodynamics option is specified, the transfer of mass and energy between the pool and the atmosphere is implicitly determined by the assumption that the pool and the atmosphere have equal temperatures. Thus, for example, if energy is deposited only in the atmosphere, the pool temperature will also rise and water vapor may be generated. Mass and energy transfers between pool and atmosphere are effectively instantaneous.

1.3.1 Solution Algorithm and Numerical Controls

If there is only one fluid field (pool or atmosphere) in a control volume, equilibrium thermodynamics must be used, even if the nonequilibrium option is specified (ICVTHR = 2 or NONEQUIL on the input record CV_THR). It must also be used to avoid numerical difficulties if either field occupies an exceedingly small fraction of the total volume. An element in the sensitivity coefficient array 4411 defines the approximate volume fraction of the smaller field below which full equilibrium will be imposed.

1.3.2 Conduction/Convection-Driven Transfer

Description of phenomena at the pool surface requires simultaneous solution of the equations of heat and mass transfer. This involves finding the temperature of the pool surface which satisfies the requirements that

(1) The evaporation or condensation mass flux is that given by the mass diffusion equation for the existing gradient in the partial pressure of water vapor between the surface and the bulk atmosphere,

(2) The net heat delivered to the interface by convection, conduction, and radiation is consistent with the latent heat required by the evaporation or condensation mass flux, and

(3) The partial pressure of water vapor at the pool surface corresponds to saturation at the surface temperature.

The heat flows from the atmosphere to the pool surface and from the pool to the pool surface are calculated from heat transfer coefficients (subject to a lower limit defined by conduction) and the bulk pool and atmosphere temperatures. Both convection and radiation are considered in evaluating the heat transfer coefficients. Energy may also be deposited "in" the surface by other packages; this option is currently used only for radiation from the COR package. The constants in the heat transfer coefficients are defined by sensitivity coefficients in C4407(*). See the CVH/FL Package Reference Manual for details.

1.3.3 Flashing Mode Mass and Energy Transfer

A source of heat within the pool or a reduction in pressure in the control volume may result in pool boiling or flashing, respectively. Water vapor appears in the pool, steam bubbles rise to the surface, and mass and energy are transferred from the pool to the atmosphere. The remaining pool may be two-phase water if all the steam bubbles have not risen to the surface.

The model in the CVH package is simple. It assumes a constant bubble rise velocity, v_B, given by sensitivity coefficient C4407(1), with a default value of 0.3 m/s. It also assumes that the steam flow is a linear function of altitude in the pool, zero at the bottom, and constant throughout a timestep. These assumptions are sufficient to calculate the mass transferred to the atmosphere during a timestep. If the resulting transfer of water vapor is not sufficient to reduce the void fraction in the pool to 0.40 or below, an additional mass of water vapor is instantaneously transferred to do so. The maximum permitted void fraction is coded as a sensitivity coefficient, C4407(11). See the CVH/FL Package Reference Manual for details.

1.4 Mass and Energy Sources and Sinks

CVH input allows the definition of explicit sources and sinks (negative sources) of mass and/or energy. Additional sources and sinks may be imposed by other packages during a MELCOR calculation; these are treated automatically and do not require input to CVH. Heat sources are included from the COR, Cavity (CAV), Fan Cooler (FCL), Fuel Dispersal Interactions (FDI), RadioNuclide (RN), Spray (SPR), and HS packages. Because a thermochemical reference point is used in thermodynamics, gas combustion modeled in the Burn (BUR) package does not involve an explicit heat source; the heat of combustion is implicit in the enthalpy functions as discussed in the CVH/FL Reference Manual. Mass sources arise from CAV and SPR; from oxidation reactions in BUR, COR, and FDI; and from condensation/evaporation, structural degassing, and melting of ice (in ice condensers) in HS.

Explicit CVH sources may be defined in terms of data from the Control Function (CF), Tabular Function (TF), and/or External Data File (EDF) packages; multiple sources may be defined in each control volume. Each mass source is associated with a specific numbered material. An energy source is associated with either the pool or the atmosphere. The association may be either direct or implicitly defined by the appropriate phase for a coupled mass source.

Sources may be defined as integrals or as rates. Rates may be either absolute or normalized per unit mass or per unit volume. For a source that changes rapidly, the user

should give serious consideration to use of the integral source (INTERP = INTEGRAL on the CV_SOU input record). With these sources, the mass or energy added during each CVH advancement is defined by differences in the running integral. The total added through any point in time is, therefore, forced to match the desired total given by the defining tabular or control function. If rates are specified, on the other hand, an interpolated value is imposed as a constant over a CVH advancement; the integral of the interpolated values may be very different from the integral of the original data.

Energy sources and sinks may be defined independently of any mass sources, but if a mass source is present in a control volume, the associated enthalpy must be included in the energy source(s) for the corresponding phase—pool or atmosphere—of that volume. (Enthalpy must be used rather than internal energy to account for volume work done in the addition process.) Energy sources may be combined if desired; there is no need to define an energy source for each mass source. Input checking is restricted to confirming that each volume containing a mass source contains at least one energy source.

In order to deposit mass with the enthalpy corresponding to a known temperature, the contribution to the energy source must be consistent with the corresponding equation of state in MELCOR. Reference points in MELCOR are defined so that all chemical heats of reaction are implicit in the enthalpy functions, as in JANAF tables. However, in order to allow use of the familiar Keenan and Keyes equation of state for water, the reference points for water, oxygen, and all other oxygen-containing species were modified from the JANAF values. See the H2O/NCG Package Reference Manual for more details.

As a result, while the user can obtain correct values for water from Keenan and Keyes steam tables, enthalpies for noncondensible gases must be calculated from data in the H2O/NCG Package Reference Manual. However, noncondensible gases in MELCOR are modeled as ideal gases, and the enthalpy of an ideal gas is a function of temperature only. The Input on the CV_SOU record allows the user to associate a temperature with a mass source. For this purpose, the mass source must immediately precede the "TE" source in the list of sources on the CV_SOU record. The tabular or control functions which define sources are associated in pairs, with the second function of each pair defining the temperature of the material added by the immediately preceding mass source function. This is sufficient to uniquely determine the enthalpy of a noncondensible gas (but not that of water). Therefore, these options are strongly recommended for use with sources of noncondensible gases.

The temperature options are much less useful for water sources and should be used with care (if used at all). Alternate methods of source definition are recommended; an example, using a time-specified volume (see Section 1.5) is given in Section 6.

If a temperature option is used for water, the definition of enthalpy is made unique by making a further assumption about the state of the added water. It is because this assumption may not correspond to the user's intentions that the temperature option is not recommended for water. If liquid water is added as the pool (material 1) or as fog (material 2) at a temperature below the critical temperature, it is assumed to be saturated at the designated temperature. If the intention was to add subcooled water, the difference can be

substantial, particularly for low temperatures and high pressures. For temperatures above critical, the enthalpy of liquid water will be calculated at critical density; the results will almost certainly not be what the user desired. For addition of water vapor (material 3), the enthalpy will be calculated at the designated temperature and a density corresponding to saturation at 0.1 MPa. Because water vapor is far from ideal, the result may differ significantly from that desired.

1.4.1 Flashing of Superheated Water Sources

MELCOR has a special model to treat the introduction of liquid water into a control volume at an elevation above the pool surface. If liquid water enters the control volume from an explicit source (specified on the CV_SOU input record), all of the mass and its associated enthalpy are simply added directly to the pool in the control volume. Water entering as fog or vapor, on the other hand, is added directly to the atmosphere. An option is available for CVH sources. If a water source is specified in this form (i.e., flashing), its partition between liquid and vapor will be internally calculated, based on equilibration (and stagnation) at the pressure of the receiving volume. If the water is superheated at the pressure of the receiving volume, the model accounts for stagnation and equilibration at that pressure.

The partition between liquid and vapor is calculated from the average enthalpy, and a fraction of the liquid is assigned to the fog field. By default, this is taken as the fraction of a Rosin-Rammler distribution that lies below a maximum diameter. If the RN1 package is active, the cut-off diameter is taken as the maximum aerosol size treated by MAEROS (DMAX on record RN1_ASP, with a default of 50 μm). If the RN1 package is not active, the maximum size is defined by a sensitivity coefficient with the same default value, 50 μm. The user may also specify the fraction directly through a sensitivity coefficient. By default, the Sauter mean diameter and exponent that characterize the Rosin-Rammler distribution is defined by sensitivity coefficients with default values of 65 μm and 5.32, respectively. The user can choose to replace this default, on a case-by-case basis, with either a different constant value or the value of a REAL control function. If, after equilibration, the mass of fog in any volume exceeds the maximum permitted, the excess is added to the pool of that volume.

If the MELCOR RN package is active, water droplets in the atmosphere are considered to behave as aerosols. In this case, the Rosin-Rammler distribution is ordinarily used to distribute the new fog over the water aerosols sections (size bins). A user option switch is provided to defeat this—in which case, the RN1 package will calculate the distribution based on the assumption that the new fog resulted from condensation. However, the resulting average droplet diameter is typically far too small, and the option should be used only for purposes of testing and demonstration.

1.5 Boundary Volumes

MELCOR hydrodynamics allows users to define thermal-hydraulic boundary conditions by specifying the state of one or more volumes as functions of time. A time-independent volume may be defined by simply setting a flag on record CV_THR and defining its state in the same manner as for a normal volume. An alternate value of this switch allows the user to define a time-specified volume, with properties that change as a function of time to follow

a history defined by tabular functions, control functions, and/or external data files. In this case, the numeric fields on the necessary records are simply replaced by references to the appropriate tabular or control functions, or to the data channel of an EDF.

Initialization of a calculation to the normal operating conditions in a power plant (or other facility) can be tedious because of the numerous mass, momentum, and energy balances involved. Time-independent volumes can be used to establish fixed boundary conditions. For example, the state of the pressurizer serves to define the system pressure while that of the steam generator secondaries is critical in defining the system temperature. Other parameters will stabilize after relatively short calculations with these volumes defined as time-independent and the results reentered as initial conditions for the transient calculation of interest.

An option is available in MELCOR 2.1, to define a volume as time-independent until a user-defined time is reached, at which time it is activated and treated thereafter as a normal volume. This allows the initialization calculation to be performed as part of the transient calculation, with the time to end initialization defined on the optional field TENDIN of the CV_TENDINI record.

The essential difference from a normal volume is that the new state in a time-independent or time-specified volume is defined in the solution of the hydrodynamic equations. Any mass and/or energy sources (in addition to masses and energies advected by CVH or transferred by other packages) required to achieve that state are computed and recorded as additional sources created in the CVH package.

These volumes may be used in any way that a normal volume may be used, including definition of conditions for vapor and aerosol phenomena in the RN package. Fission products in the RN package are transported with material flows into and out of normal control volumes. They are treated as infinitely dilute in time-independent and time-specified volumes and are, therefore, transported into, but not out of, volumes. This is necessary to prevent fission products from being drawn back into a failed containment from the environment (modeled as a time-independent volume) if containment pressure falls below atmospheric pressure.

A time-specified volume may also be used to define the enthalpy of mass sources and is particularly useful for water sources. See Section 1.4 and the example in Section 6.

1.5.1 Volume Velocities for Boundary Volumes

The characteristic volume velocity may be needed to calculate an appropriate heat transfer coefficient in an initialization or boundary volume. For an initialization volume (ICVACT = -3), this velocity is calculated even before the volume becomes active at time TENDIN. The internal calculation for normal volumes is used, based on the default flow area (volume/height) if no area is input on a CV_ARE record. For a time-independent or time-specified boundary volume (ICVACT = -1 or -2) the velocity will be calculated if and only if an area is input on a CV_ARE record. The normal internal calculation will be used.

The CV_ARE record may also be used to specify a control function to define the velocity. This option is expected to be most useful for time-independent or time-specified boundary volumes, but is available for any volume.

2 User Input

CVH input to MELGEN defines the initial state of control volumes. There is no provision to modify conditions on restart; CVH input to MELCOR is limited to activation of additional optional output, which may be useful in identifying the source of calculational difficulties.

Several input options that were included with the comment, "Do not use" in the previous versions of users' guides are removed.

2.1 MELGEN User Input

The user input for the CVH package is described below. The following set of input (CV_XXX records) is required and is used to describe each individual control volume defined for a given application. An arbitrary number of control volumes may be defined. Practically, the number is limited only by the available memory on the computer.

CV_DFT – CVH Package Default Scheme Record

Optional.

This record modifies a set of sensitivity coefficient values in the CVH package to the version 1.8.6 standards. Note that these values may not reflect current MELCOR best practice modeling parameters for the analysis of severe accidents. If the record is present, one floating point field must be present. This record can be specified once and is not needed for each control volume (i.e. the record is not associated with any particular control volume record accompanied by CV_ID).

(1) DEFAULT

Mode of default scheme. Permitted values are 1.86 and 2.0.

(type = real, default = 2.0)

If the DEFAULT field is 2.0, no change is made to the existing sensitivity coefficients. If the DEFAULT field is 1.86, the following CVH package sensitivity coefficients will be modified, unless otherwise specified by the user input record.

Sensitivity Coefficient	DEFAULT = 1.86	DEFAULT = 2.0
4413 (5)	1.0E-6	0.05
4414 (1)	0.001	0.01
4415 (1)	0.5	1.0

Examples

```
CV_DFT  1.86
```

CV_TENDINI – Time to end initialization

Optional

The ICVACT field on record CV_THR may be used to declare volumes as time-independent for an initialization segment of a calculation to allow stabilization of normal operating conditions in a power plant or other facility. With ICVACT = -3 or DELAYED-ACTIVE, this record defines the end of the initialization period; if absent, any volume so declared will be activated at the start of the MELCOR calculation. See discussion in Section 1.5.

If ICVACT = -3 or DELAYED-ACTIVEon CV_THR record then

(1) TENDIN

Time to end initialization and activate all volumes with ICVACT = -3.

(type = real, default = TSTART, units = s)

CVH_ATMCS – Atmosphere Sound Speed Model Selection

Optional

Added capability to use modified definition of sonic velocity in atmosphere for comparison with CONTAIN code (see CVH Reference Manual for more details). This record has no effect on calculations unless keyword "FMOD" is used in this record entry.

(1) AtmOption

Keyword for input value

(a) 0 or DEFAULT

FMOD is not active

(b) 1 or FEM

FMOD is invoked

(type = character*7, default = DEFAULT)

Example

```
CVH_ATMCS    FMOD
```

CVH_CSTBL – Print option to output a special table for the sound speed studies

Optional

Added option to output a table on sound speed information .

(1) WriteAtmTable

Keyword for input value:

(a) YES
Table is output

(b) NO
No table is output

(type = character*3, default = NO)

Example

```
CVH_CSTBL    YES
```

CV_ID – Control Volume Name and Sequence Number

Required.

This record defines the unique name of the control volume and optionally a user-defined sequence number of the control volume. The second field, ICVNUM, is optional and may be omitted.

(1) CVNAME

User-defined control volume name.

(type = character*16, default = none)

(2) ICVNUM

User-defined control volume sequence number.
Optional.
If omitted, a sequence number will be assigned by incrementing the last sequence number by 10.

(type = integer, default = none)

Examples

```
CV_ID  'VOLUME ONE'  110
```

CV_DELETE – Ignore all records associated with CV_ID

Optional

This record is processed after all records in the input file are read. It will turn off all CV records related to the CVNAME identifier.

(1) CVNAME

User-defined control volume name.

(type = character*16, default = none, units = none)

Example

```
CV_DELETE  'VOLUME ONE'
```

CV_TYP – Control Volume Type

Optional.

This record defines the control volume type.

(1) ICVTYP

Type of control volume.
This entry is used to define groups of associated control volumes for editing and accounting purposes only (currently only the RN package editing/accounting routine uses this flag to permit editing by CV type; see the RN Package Users' Guide for more information). For example, each control volume associated with the reactor coolant system might be assigned ICVTYP = RCS. One control type, TYPE99, is reserved for RN package FILTERS by default.

(type = character*16, default = none)

Examples

```
CV_TYP   CVType01
CV_TYP   'RCS'
```

CV_THR – Switches for Thermodynamic Input

Required.

This record defines the attributes of the control volume.

The user can also specify that no fog may exist in the control volume atmosphere. In this case, all water will be put into the pool.

A volume may also be declared time-independent or time-specified. A volume may further be declared to be time-independent only during an initialization segment of a calculation and become a normal volume thereafter. See Section 1.5. The time to end initialization is defined on the additional optional field TENDIN.

(1) ICVTHR

Control volume thermodynamics switch. The nonequilibrium option is recommended in general.

(a) 1 or EQUIL
Equilibrium (Tpool = Tatmos).

(b) 2 or NONEQUIL
Nonequilibrium (Tpool ≠ Tatmos).

(type = integer / character*8 , default = none)

(2) IPFSW

Fog/no fog switch.

(a) 0 or FOG
Fog allowed (default).

(b) 2 or NOFOG
No fog (any liquid water is moved to pool).

(type = integer / character*5, default = none)

(3) ICVACT

Active/inactive switch.

(a) 0 or ACTIVE
Active, with state advanced by integrating the conservation equations, (default).

(b) -1 or TIME-INDEP
Time-independent.

(c) -2 or PROP-SPECIFIED
Properties specified as a function of time.

(d) -3 or DELAYED-ACTIVE
Treated as time-independent (but with a velocity calculation performed by default) for TIME < TENDIN, and as active for TIME ≥ TENDIN. TENDIN may be input on optional field below.

(type = integer / character*14, default = 0 or active, units = dimensionless)

Example

```
CV_THR NonEquil FOG ACTIVE              !ICVTHR, IPFSW, ICVACT
CV_THR NonEquil FOG DELAYED-ACTIVE 100.0
```

CV_VEL – Control Volume Initial Velocities

Optional.

Average velocities in the control volume atmosphere and pool are used by the CVH, COR, and HS packages in calculation of heat transfer coefficients. This record may be used to define nonzero initial values of these velocities, which will be used until internally calculated values become available during the first advancement in MELCOR.

(1) VLATMO

Velocity of the atmosphere.

(type = real, default = 0.0, units = m/s)

(2) VLPOLO

Velocity of the pool.

(type = real, default = 0.0, units = m/s)

Example

```
CV_VEL  0.23  0.56
```

CV_ARE – Control Volume Area for Velocity Calculation

Optional.

During advancement, the velocities in each control volume are internally calculated from the area of the control volume and the area, void fraction, and velocities in the flow paths connected to that control volume. The default value for the control volume area is given by its volume divided by its height for normal volumes with ICVACT = 0 or –3.

For time-independent or time-specified volumes (ICVACT = -1 or -2), the default is to bypass the calculation and assume that velocities are zero. However, in MELCOR 1.8.6, the normal internal calculation will be performed if an area is input. The result will, of course, be inappropriate unless all flow connections to the volume are included in the MELCOR nodalization. (Note that volume sources are not treated as flow connections.)

The user may also define the velocity directly by specification of a control function that provides the value (in m/s). This option is available for any volume, but is expected to be particularly useful for time-independent or time-specified boundary volumes.

(1) ICFVEL

(a) 0 or NOCF

Control function is not used.

(b) -1 or CF

Use control function.

(type = integer/character*4, default = none, units = none)

followed by

(2) CVARA

Control volume area used for velocity calculation.
Input if ICFVEL = 0 or NOCF.

(type = real, default = volume/height, units = m^2)

or

(2) NCFVEL

User name of control function that defines the characteristic fluid velocity in this volume.
Input if ICFVEL = -1 or CF.

(type = character*16, default = none, units = none)

Examples

```
CV_ARE NOCF   10.0  ! ICFVEL, CVARA
CV_ARE CF     CF101 ! ICFVEL, NCFVEL
CV_ARE 0      0.9   ! no CF is used, CV area
```

2.1.1 Thermodynamic input

The user must provide sufficient information to define the initial thermodynamic state of the control volume. The fundamental internal variables in MELCOR are mass and internal energy, but options are provided that allow the user to specify the state in terms of more commonly available quantities. If these options are employed, the user-input quantities are used to calculate mass and energy. These are then used as arguments to the equation of state in the CVT package, which may not reproduce input pressures and temperatures exactly. In addition, if nonequilibrium conditions (unequal pool and atmosphere temperatures) are input for a volume in which equilibrium thermodynamics is specified (ICVTHR = 1 or EQUIL on record CV_THR), the input will be processed and the resulting (nonequilibrium) contents equilibrated. Two input options are available:

(1) Pressures, temperatures, and mass fractions input. This option is retained for compatibility with MELCOR versions prior to 1.7.0.

Input pressures and/or temperatures. Specify the location of water as mass fractions in the pool, fog, and/or atmosphere. Specify the composition of the noncondensible gas in the atmosphere as mass fractions. Actual amounts of all materials are then inferred from pressures and temperatures.

(2) Separate pool and atmosphere input. This option is recommended for normal applications. It is the only option permitted for time-specified volumes (ICVACT = -2 on the CV_THR record).

Input pressures and/or temperatures. Specify the amount of pool as a mass, volume, or surface elevation. Specify the composition of the atmosphere in terms of its water content (partial pressure, relative humidity, or dew point temperature)

and the mole fractions of noncondensible gases. An initial void fraction in the pool and/or fog content in the atmosphere may also be specified.

CV_PAS – Pool and Atmosphere Switches

Required.

This record is used to specify the presence of a pool and/or an atmosphere in each control volume.

(1) ITYPTH

Type of thermodynamic input.

(a) 2 or COMMON
Pressures, temperatures, and mass fractions.

(b) 3 or SEPARATE
Separate pool and atmosphere input.

(type = integer / character*8, default = 3, units = dimensionless)

(2) IPORA

The key setting if a pool, an atmosphere, or both pool and atmosphere are present.

(a) 1 or ONLYPOOL
Only pool is present.

(b) 2 or ONLYATM
Only atmosphere is present.

(c) 3 or POOLANDATM
Both pool and atmosphere are present.

(type = integer / character*10, default = none)

If IPORA = 1 or ONLYPOOL, then the WATERSTATE field is required, and the input of VAPORSTATE is not permitted (an error message will be issued). Similarly, if IPORA = 2 or ONLYATM, then the VAPORSTATE field is required, and the input of WATERSTATE is not permitted (an error message will be issued). If both pool and atmosphere are present, then both keys are required.

If a pool is present (IPORA = 1 or 3), specify its thermodynamic state:

(3) WATERSTATE

The key setting whether the water is subcooled or saturated.

(a) 0 or SATURATED
(b) 1 or SUBCOOLED

(type = integer / character*9, default = none)

If an atmosphere is present (IPORA = 2 or 3), specify its thermodynamic state

or

(3) VAPORSTATE

The key setting whether the water vapor is superheated or saturated.

(a) 0 or SATURATED
(b) 1 or SUPERHEATED

(type = integer / character*11, default = none)

Examples

```
CV_PAS 2      ONLYATM SUPERHEATED
CV_PAS COMMON OnlyPool SUBCOOLED
CV_PAS 3      PoolAndAtm SATURATED SUPERHEATED ! separate
```

The input variables required and the treatment of defaults are different for each type of thermodynamic input (ITYPTH), though the record names are the same. The input for time-independent volumes is the same as for normal volumes; that for time-specified volumes is discussed following the description of input for ITYPTH = 3.

Mass Fraction Input (ITYPTH = 2)

Input the total volume pressure, the partial pressure of water vapor in the atmosphere, and the temperatures of the pool and the atmosphere. Specify the partition of water among pool, atmosphere, and fog as water mass fractions; the actual masses are implicitly defined by the temperatures, the pressures, and the total volume. (If both water vapor and fog are present, the partial pressure of water in the atmosphere must correspond to saturation at the atmosphere temperature.)

Specify the composition of the noncondensible gas in the atmosphere as noncondensible gas mass fractions; the actual amounts are implicitly defined by the difference between the partial pressure of water vapor in the atmosphere and the total pressure. The control volume pressure is defined as the pressure at the pool/atmosphere interface. If no pool is present, it is the pressure at the lowest elevation in the cell. If no atmosphere is present, it is the pressure at the highest elevation of the cell.

CV_PTD – Control Volume Pressure or Temperature Data

Required.

The user must specify the thermodynamic state of the pool and/or the atmosphere (pressure and temperature). For ITYPTH = 2, this record defines the total volume pressure. The temperatures of the pool and/or of the atmosphere are defined on the records below.

(1) PTDID

Keyword for input value.
PVOL Control volume pressure.

(type = character*4, default = none)

followed by

(2) PVOL

Control volume pressure (pool pressure if a pool is present, and total pressure of water vapor in the atmosphere if there is no pool and only an atmosphere is present).

(type = real, default = none, units = Pa)

Example

```
CV_PTD PVOL   7.0E+06   ! pressure, PVOL
```

CV_PAD – Pool Additional Data

Required if SUBCOOLED pool is present. This record defines the pool temperature.

The record must be absent if there is no pool or if the pool is saturated.

(1) TPOL

Temperature of pool.

(type = real, default = none, units = K)

Example

```
CV_PAD    300.  ! TPOL
```

CV_AAD – Atmosphere Additional Data

Required if a SUPERHEATED atmosphere is present. This record defines the atmosphere temperature.

The record must be absent if there is no atmosphere or if the atmosphere is saturated.

(1) ATMID

Keyword for input value.

(a) TATM

Temperature of atmosphere.

(type = character*4, default = none)

followed by

(2) TATM

Temperature of atmosphere.

(type = real, default = none, units = K)

Example

```
CV_AAD    TATM   500. (CVolume(i).TEMPSO(2))
```

CV_BND – Boundary between the Pool and the Atmosphere

Required.

(1) BNDID

Keyword for input value.

(a) FRAC

Mass fractions for materials POOL, FOG, and H2O VAP. If water is present, the mass fractions for materials POOL, FOG, and H2O-VAP must sum to 1.0.

(type = character*4, default = none)

followed by three required fields:

(2) PMFRC

Fraction mass of POOL.

(type = real, default = none, units = kg)

(3) FMFRC

Fraction mass of FOG.

(type = real, default = none, units = kg)

(4) VMFRC

Fraction mass of H2O-VAP.

(type = real, default = none, units = kg)

Example

```
CV_BND FRAC 0.5 0.2 0.3
```

Separate Pool and Atmosphere Input (ITYPTH = 3)

CV_THERM – Alternative Input for Thermodynamic Data

Required unless alternative input below is used.

This new input is the preferred input format for the thermodynamic data of the control volume. It will eventually replace the alternative input records below. When this input is used, the following input cards should not be entered: CV_BND, CV_PTD, CV_VOID, CV_AAD, CV_NCG. Note that the time-specified volumes are still entered through CV_BND input card. This new input is only used with ITYPTH = 3.

(1) NROW

Number of rows in the table – data pairs are expected

(type = integer, default = none, units = dimensionless)

The total pressure in the volume applies to the pool and/or the atmosphere. It may be specified either directly or as corresponding to saturation at some desired temperature. The latter option is useful for situations involving only saturated water. The data pair is expected. Either PVOL or TSAT must be specified.

The following guidance should be used to develop the following table. Note the data pair is expected–a string is followed by a value.

It is recommended to enter the pressure of the volume or its saturation temperature.

(1) PVOL

Control volume pressure

(type = real, default = None, units = Pa)

or

(1) TSAT

Saturation temperature at control volume pressure

(type = real, default = None, units = K)

If a pool is present, enters one of three input keywords/values for the pool first. Then enter other parameters that describe the pool characteristics.

(2) MPOL

Mass of pool

(type = real, default = 0.0, units = kg)

or

(2) VPOL

Volume of the pool

(type = real, default = 0.0, units = m^3)

or

(2) ZPOL

Elevation of the pool surface

(type = real, default = none, units = m)

(3) TPOL

Temperature of pool.

If zero, negative, or not input, the temperature of the pool is set to saturation at the pool pressure (PVOL or PSAT(TSAT))

(4) VOID

Void fraction in the pool, in the range 0.0 ≤ VOID < 1.0. Because noncondensibles may not reside in the pool, any bubbles must be water vapor. Therefore, the input of VOID is permitted only for a saturated pool. If not input, the pool void fraction will be zero.

(type = real, default = 0.0, units = dimensionless)

If an atmosphere is present and saturated, the partial pressure of water vapor (PH2O) must equal to the saturation pressure (PSAT). Specify PH2O in the atmosphere directly or by the relative humidity (RHUM) or the dewpoint temperature (TDEW). Note that input of a non-negative value for PH2O or of a value for either RHUM or TDEW fully defines the water vapor content of the atmosphere. Over specification (by input of more than one of these) is not permitted.

PH2O = RHUM x saturation pressure (PSAT) evaluated at atmosphere temperature (TATM)

or

PH2O = PSAT(TDEW)

If the atmosphere is superheated, specify only PH2O or the temperature if it is saturated.

(5) PH2O

Partial pressure of water vapor in the atmosphere. If negative or not input, this partial pressure will be set according to the equations above. Note that if PH2O is not specified from TDEW or RHUM, then PH2O is set at PSAT(TATM) or PH2O = PVOL if TATM is not specified.

(type = real, default = -1.0, units = Pa)

or

(5) RHUM

Relative humidity of the atmosphere, in the range 0.0 ≤ RHUM ≤ 1.0 (see first equation above).

(type = real, default = 1.0, units = dimensionless)

or

(5) TDEW

Dewpoint temperature in the atmosphere. The dewpoint temperature is defined as the saturation temperature to the partial pressure of water vapor (see second equation above) (type = real, default = none, units = K)

(6) TATM

Temperature of atmosphere. If zero, negative, or not input, the saturation temperature at PH2O will be used. Note if RHUM is specified, then TATM input must be non-zero and positive.

(type = real, default = -1.0, units = K)

If noncondensible gas is present in the atmosphere, specify its composition as mole fractions. The actual amounts of the various gases are implicitly defined by the difference between the partial pressure of water vapor (PH2O) in the atmosphere and the total pressure (PVOL or PSAT(TSAT)). If this difference is zero, input for noncondensible gas is not permitted. To input condensible gases (NCG), the name of the NCG (from the list of NCG materials) is a keyword, then it is followed by the value of the mole fraction. Data pair is expected. Note the sum of the mole fractions must be added to 1.0.

(7a) NCG

Noncondensible gas name

(type = character*8, default = none)

(7b) mlfr

Mole fraction of noncondensible gas (NCG).

(type = real, default = none, units = dimensionless)

If an atmosphere is present and the partial pressure of water vapor corresponds to saturation, optionally specify the fog content either directly by its mass or indirectly by the volume fraction.

(8) MFOG

Mass of fog

(type = real, default = 0.0, units = kg)

or

(8) VFFOG

Volume fraction of fog in the atmosphere in the range 0.0 ≤ VFOG < 1.0.

(type = real, default = 0.0 units = dimensionless)

Example

```
CV_THERM   4
           1 PVOL 2.0E05
           2  ZPOL  10.0  TPOL 320.0  VOID 0.0
           3  RHUM 0.2  TATM 380.0
           4  N2 0.8   O2 0.2
```

CV_PTD – Control Volume Pressure or Temperature Data

Required.

The user must specify the thermodynamic state of the pool and/or the atmosphere (pressure and temperature). This record allows the definition of control volume pressure or the corresponding saturation temperature. Additional parameters for the pool and/or the atmosphere are described on records below.

(1) PTDID

Keyword for input value.

(a) PVOL

Control volume pressure.

(b) TSAP

Saturation temperature of pool.

(c) TATM

Temperature of atmosphere.

(type = character*4, default = none)

optionally followed by

(2) PVOL

Control volume pressure (pool pressure if a pool is present, and total pressure of water vapor in the atmosphere if there is no pool and only an atmosphere is present).

(type = real, default = none, units = Pa)

or

(2) TSAP

Saturation temperature of the pool, used to define the pressure (which is also the total control volume pressure).

(type = real, default = none, units = K)

or

(2) TATM

Temperature of atmosphere.

(type = real, default = none, units = K)

Example

```
CV_PTD PVOL   7.0E+06   ! pressure ,PVOL
```

Pool Input:

If a pool is present, the user must specify its thermodynamic state. If a pool is saturated, its thermodynamic state is completely specified by pressure or saturation temperature input on the CV_PTD record. An initial void fraction may also be defined for a saturated pool on the CV_VOID record. If it is subcooled, the user must specify the additional parameter on the CV_PAD record.

CV_PAD – Pool Additional Data

Required if a subcooled pool is present.

If TSAP or TATM on the CV_PTD record are defined, then the additional temperature will be used to define the pool pressure.

(1) TPOL

Temperature of the pool.

(type = real, default = none, units = K)

Example

```
CV_PAD  320.0   !TPOL
```

CV_VOID – Void Fraction in the Pool

Optional.

If the pool is saturated, an initial void fraction may also be defined.

(1) VOID

Void fraction in the pool, in the range $0.0 \leq \text{VOID} < 1.0$. Because noncondensible gases may not reside in the pool, any bubbles must be water vapor. Therefore, input of VOID is permitted only for a saturated pool. If not input, the pool void fraction will be zero.

(type = real, default = 0.0, units = dimensionless)

Example

```
CV_VOID 0.1
```

Atmosphere Input:

If a pool is present, the atmosphere, including fog if present, occupies the remainder of the control volume. If the pool fills the volume, no additional atmosphere properties are needed.

If an atmosphere is present, its thermodynamic state is defined by the volume pressure input on the CV_PTD record and one or two additional parameters input on the CV_AAD and CV_NCG records. If the atmosphere is superheated and noncondensible gas is not present in the atmosphere, the temperature (TATM, greater than saturation for the given vapor pressure) or the partial pressure of water vapor in the atmosphere (by the relative humidity RHUM on the CV_AAD record) must be specified. If the atmosphere is superheated and noncondensible gas is present in the atmosphere, any pair of two parameters (for the atmosphere on the CV_AAD record and for NCG on the CV_NCG record) must be specified. If the atmosphere is saturated, the CV_NCG record is required only if noncondensible gas is present in the atmosphere.

CV_AAD – Atmosphere Additional Data

Optional.

(1) ATMID

Keyword for input value.

(a) TATM

Temperature of atmosphere.

(b) RHUM

Relative humidity of the atmosphere.

(type = character*4, default = none)

optionally followed by

(2) TATM

Temperature of atmosphere.

(type = real, default = none, units = K)

or

(2) RHUM

Relative humidity of the atmosphere, in the range $0.0 \leq RHUM \leq 1.0$. Relative humidity is defined as the ratio of the partial pressure of water vapor to the saturation pressure at the atmosphere temperature.

(type = real, default = none, units = dimensionless)

Example

```
CV_AAD TATM   300.0
```

CV_NCG – NCG Data Table

Optional.

If noncondensible gases are present in the atmosphere, specify the additional parameters for the thermodynamic state and composition as mole fractions. The actual amounts of the various gases are implicitly defined by the difference between the partial pressure of water vapor in the atmosphere and the total pressure. If this difference is zero, no input is permitted.

(1) NMMAT

Number of NCG materials.

(type = integer, default = 0)

(2) NCGID

Keyword for input value.

(a) PNCG

Partial pressure of noncondensible gases in the atmosphere.

(b) PH2O

Partial pressure of water vapor in the atmosphere.

(c) TDEW

Dew point temperature in the atmosphere.

(d) RHUM

Relative humidity of the atmosphere.

(e) TATM

Temperature of atmosphere.

(f) SAT

Required input for certain cases. See input descriptions for CV_BND

followed by

(3) PNCG

Partial pressure of noncondensible gases in the atmosphere.

(type = real, default = none, units = Pa)

or

(3) PH2O

Partial pressure of water vapor in the atmosphere.

(type = real, default = none, units = Pa)

or

(3) TDEW

Dew point temperature in the atmosphere. The dew point temperature is defined as the saturation temperature corresponding to the partial pressure of water vapor.

(type = real, default = none, units = K)

or,

(4) RHUM

Relative humidity of the atmosphere

(type = real, default = none, units = dimensionless)

or,

(5) TATM

Temperature of the atmosphere

(type = real, default = none, units = K)

The following data are input as a table with length NMMAT:

(1) NUM

Table row index.

(type = integer, default = none)

(2) NAMGAS

Noncondensible gas name (from the list of NCG materials).

(type = character*8, default = none)

(3) MLFR

Mole fraction of noncondensible gas NAMGAS.
If noncondensible gases are present, the sum for all NCG materials should be equal to 1.0.

(type = real, default = none, units = dimensionless)

Example

```
CV_NCG 2 RHUM 0.9 !N  NAMGAS  MLFR
                  1    N2     0.8
                  2    O2     0.2
```

CV_FOG – Fog Data

Optional.

If an atmosphere is present, and the partial pressure of water vapor corresponds to saturation for its temperature. This record is used to specify the fog content either directly by its mass or indirectly by the volume fraction in the total atmosphere (but not both).

(1) FOGPARAM

Keyword for input value:

(a) 0 or MASS
Mass of fog.

(b) 1 or FRAC
Volume fraction of fog.

(type = character*4, default = none)

followed by

(2) MASS

Mass of fog.

(type = real, default = none, units = kg)

or

(2) VFOG

Volume fraction of fog in the atmosphere in the range $0.0 \leq \text{VFOG} < 1.0$.

(type = real, default = none, units = dimensionless)

Example

```
CV_FOG   FRAC   0.2
```

CV_BND – Boundary between the Pool and the Atmosphere

Required if both a pool and an atmosphere are present.

(1) BNDID

Keyword for input value:

(a) MASS

Mass of pool.

(b) VPOL

Volume of the pool.

(c) ZPOL

Elevation of the pool surface.

(type = character*4, default = none)

followed by

(2) MASS

Mass of pool.

(type = real, default = none, units = kg)

or

(2) VPOL

Volume of the pool.

(type = real, default = none, units = m^3)

or

(2) ZPOL

Elevation of the pool surface.

(type = real, default = none, units = m)

Example

```
CV_BND ZPOL   9.9
```

<u>Input for Time-Specified Volumes</u>

The properties of a volume may be specified as user-defined functions of time by setting ICVACT = -2 on record CV_THR. The permitted input variables are a subset of those described above (records CV_PTD, CV_BND, CV_PAD, CV_VOID, CV_AAD, and CV_NCG). These variables (e.g., PVOL) are used, but with a function or file reference replacing the constant value. Permitted references are TF, CF, or EDF, denoting a tabular function, a control function, or a channel of an external data file, respectively. One or two additional fields are required to define the name of the function or EDF file and the number of the EDF data channel.

(1) CFTFSELECT

Switch for variable data:

(a) -1 or CF
Control function.

(b) 1 or TF
Tabular function.

(c) 2 or EDF
External data file.

(type = integer / character*3, default = none)

optionally followed by

(2) CFNAME
Control function name (if CFTFSELECT = CF or -1).

(type = character*16, default = none)

or

(2) TFNAME
Tabular function name (if CFTFSELECT = TF or 1).

(type = character*16, default = none)

or

(2) EDFNAM
External data file name (if CFTFSELECT = EDF or 2).

(type = character*16, default = none)

and

(3) NUMCHN
Channel number containing the data.

The tabular function, control function, or external data file should define the variable as a function of time in the appropriate SI units. Keywords that are currently recognized are PVOL (on record CV_PTD), ZPOL (on record CV_BND), TPOL (on record CV_PAD), VOID (on record CV_VOID), TATM (on record CV_AAD), PH2O (on record CV_NCG), and MLFR (on record CV_NCG). They are interpreted as follows:

(a) PVOL
Control volume pressure. Required.
(type = real, default = none, unit = Pa)

(b) ZPOL
Elevation of pool surface. Optional. <u>If absent, no pool will ever be present</u>, and input of TPOL or VOID is not

permitted.
(type = real, default = none, unit = m)

(c) TPOL
Temperature of pool. Optional, permitted only if ZPOL is input. Note that TPOL must be specified if a subcooled pool will ever be present. During periods when there is no pool (as specified by ZPOL), the value of TPOL will not be used. When a pool is present, a value of TPOL equal to or less than C4409(6) (default, 1.0 K) above saturation of PVOL will be interpreted as saturation. Any greater superheat will result in termination of the calculation. Note that MELCOR thermodynamics does not permit a superheated pool; the tolerance on input is intended only to accommodate minor discrepancies resulting from interpolation in tabular data when a saturated pool is intended.
If ZPOL is input and TPOL is not, the pool (if present) will be saturated at PVOL.
(type = real, default = -1.0, unit = K)

(d) VOID
Void fraction in the pool. Optional, permitted only if ZPOL is input. Ignored unless the pool is saturated. If not input, the pool void fraction will be zero.
(type = real, default = 0.0, unit = dimensionless)

(e) TATM
Temperature of atmosphere. Optional. If an atmosphere is present and TATM is not input, the atmosphere will consist of saturated water vapor at PVOL. Note that TATM must be specified if the atmosphere will ever be superheated or contain noncondensible gases. It is impossible for input checking to assure that PVOL and TATM will always specify a superheated state. Therefore, if TATM is specified, the composition of noncondensibles (MLFR) must also be specified, see below.
(type = real, default = 1.0, unit = K)

(f) PH2O
Partial pressure of water vapor in the atmosphere. Optional, currently permitted only if TATM is input. In execution, a value of PH2O corresponding to saturation at TATM or at a temperature equal to or less than C4409(6) (default, 1.0 K) below TATM will be interpreted as saturation at TATM. Any greater subcooling will result in termination of the calculation. Note that MELCOR thermodynamics does not permit a

supersaturated atmosphere. As discussed for keyword TPOL, the tolerance on input is intended only to allow for imprecision in interpolation. If TATM is input and PH2O is not, the partial pressure will be the minimum of PVOL and saturation at TATM (in this case keyword SAT must present in CV_NCG record).
(type = real, default = 1.0, unit = Pa)

(g) MLFR
Mole fraction of noncondensible gas NAMGAS. Required if TATM is input, not permitted otherwise. (If TATM is input, both P and T will be specified.) If input, the values must sum to 1.0 within C4409(5) (default, .001); the tolerance allows for some imprecision in interpolation of tabular data.
(type = real, default = 0.0, unit = dimensionless)

In this example, channel 3 of external data file "EDF101" should contain the desired pressure in Pascals (Pa).

Example

```
CV_PTD PVOL EDF EDF101 1
CV_AAD TATM EDF EDF101 2
CV_NCG 2 PH2O EDF EDF101 3
           1 'N2' EDF EDF101 4
           2 'O2' EDF EDF101 5
```

CV_VAT – Altitude/Volume Table

Required.

The input data are altitude-hydrodynamic volume pairs. The data are input as a table. The first element of a pair is the altitude and the second is the volume initially available to be occupied by material in the CVH package (free volume). Virtual volume is not included in these input data. At least two pairs must be input. The volume corresponding to the lowest elevation must be zero. For other data pairs, if the input volume is positive, it is interpreted as the total volume between the bottom (lowest elevation input) and the current input elevation. If it is negative, the absolute value of the volume is interpreted as the volume change from the next-lower input elevation.

(1) ICVVZP
Number of control volume pairs.
(type = integer, default = none)

The following data are input as a table with length ICVVZP:

(1) NCVZ

Table row index.

(type = integer, default = none)

(2) CVZ

Altitude of current control volume pair.

(type = real, default = none, units = m)

(3) CVVOL

Volume initially available to be occupied by material in the CVH package (free volume).

(type = real, default = none, units = m^3)

Example

```
CV_VAT   5   !N   CVZ      CVVOL
              1   6.0      0.00
              2   7.0     -6.75
              3   8.0     -6.75
              4   9.0     -6.75
              5   10.0    -6.75
```

CV_SOU – External Mass and Energy Sources

Optional.

These records are used to define user input mass and energy (enthalpy) sources and sinks in addition to those calculated by other packages. As discussed in Section 1.4, such sources may be prescribed using control functions, tabular functions, or external data files. All data should be in correct SI units: J, kg, W, kg/s, K, etc.

The data are input as a table with dimension equal to the total number of external sources. Each line of the table contains the data for one source. Each data string consists of an identifier field (source index), a character switch specifying the type of source, and optional additional fields.

Mass and energy sources may, in general, be defined independently as long as the enthalpy associated with each mass source is properly accounted for. However, a "TE" type source defines an energy source in terms of the temperature of an associated "MASS" source. This mass source must immediately precede the "TE" source in the list of sources on this record. If this requirement is not met and there is no "MASS" source record immediately preceding a "TE" source record, MELGEN will trap the error, write a diagnostic, and not write a cycle zero restart.

(1) IESPNT

Number of external mass or energy sources.

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length IESPNT:

(1) NSRC

Table row index.

(type = integer, default = none, units = dimensionless)

(2) CTYP

Character field defining mass or energy source.

(a) FILE

Vector of mass and enthalpy sources, obtained from an external data file. The optional FL_EDF record can be used to record the mass and enthalpy flows through a flow path.

(b) AE

External enthalpy source for the atmosphere.

(c) PE

External enthalpy source for the pool

(d) MASS

Mass source for material.

(e) TE

Specific enthalpy of the previous mass source is defined as a function of temperature. Temperature is defined by the tabular function or control function. An atmosphere or pool destination is determined by the mass source.

(f) WM

Total mass associated with water source, to be partitioned among pool, fog, and vapor according to the flashing model described above and controlled by sensitivity coefficient array 4500.

(g) WE

Total enthalpy associated with water source, to be partitioned among pool, fog, and vapor according to the flashing model described above and controlled by sensitivity coefficient array 4500.

(type = character*4, default = none)

Table 2.1 summarizes the different fields required depending on CTYP and is followed by a description of the fields. The number of the field is omitted in the description.

If CTYP = FILE

(1) EDFNAM

Name of external data file. This file must contain at least NUMMAT+2 channels (where NUMMAT is the number of materials in the problem). Of these, the first NUMMAT+2 channels will be used as data for NUMMAT NCG masses and energies of pool and atmosphere.

(type = character*16, default = none)

(2) ESSCAL

Scale factor for data from EDFNAM.

(type = real, default = 1.0, units = dimensionless)

If CTYP = AE, PE, WM, MASS, TE and WE

( ) INTERP

Interpretation of source data.

(a) RATE
Data are to be interpreted as rates (W or kg/s).

(b) INTEGRAL
Data are to be interpreted as cumulative (J or kg).

(c) RATE/MASS
Data are to be interpreted as rates per unit mass.

(d) RATE/VOL
Data are to be interpreted as rates per unit volume.

(type = character*9, default = none)

( ) IESSRC

Input switch defining location of source data.

(a) -1 or CF
Data from control function CFNAME input below.

(b) 1 or TF
Data from tabular function TFNAME input below.

(c) 2 or EDF
Data from channel NUMCHN of external data file EDFNAM input below.

(type = integer / character*4, default = none)

followed by

( ) CFNAME

The control function name associated with the field IESSRC.

(type = character*16, default = none)

or

( ) TFNAME

The tabular function name associated with the field IESSRC.

(type = character*16, default = none)

or

( ) EDFNAM

Name of external data file associated with the field IESSRC.

(type = character*16, default = none)

( ) NUMCHN

Channel number.

(type = integer, default = none, units = dimensionless)

( ) ESSCAL

Scale factor for data from IESSRC.

(type = real, default = 1.0, units = dimensionless)

( ) IDMAT

Material name or number for mass source:

(a) 1 or POOL
Pool.

(b) 2 or FOG
Fog.

(c) 3 or H2O-VAP
Atmospheric vapor.

(d) n ≥ 4 or NCG name
A noncondensible gas identified by the input to the NCG package. Note: For TE source only, this value is required as a placeholder for the present. The value from the immediately-preceding mass source will be used.

(type = integer/character(8), default = none, units = dimensionless)

( ) ELEV

Elevation of the source in the volume.

(type = real, default = none)

( ) ISAUTOPT

Input switch defining Sauter mean diameter of water droplets.

(a) CONST
Use constant value DIAM.

(b) CF

Use control function CFDIAM.

(c) SC
Use value of sensitivity coefficient C4500(3).

(type = character*5, default = none)

optionally followed by

( ) DIAM
Constant value for Sauter mean diameter of water droplets.

(type = real, default = 0.0)

or

( ) CFDIAM
Control function name which defines Sauter mean diameter of water droplets.

(type = character*16, default = none)

( ) IFOGOPT
Fog option flag. , ignored if RN is inactive

(d) =0:
Distribute fog over RN sections directly.

(e) ≠0:
Allow RN to calculate distribution, assuming condensation.

(type = integer, default = 0)

Example

```
CV_SOU   2   !N CTYP INTERP IESSRC CF/TFNAME IDMAT ESSCAL
              1 MASS RATE   CF     CF498     POOL  1.0
              2 TE   RATE   CF     CF499     -1    1.0
```

Table 2.1 Input Fields for Each Source Type.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
NSRC	CTYP:									
	├ FILE	EDFNAM	ESSCAL							
	├ AE ├ PE ├ WM	INTERP	IESSRC:							
			├ CF	CFNAME	ESSCAL					
			├ TF	TFNAME	ESSCAL					
			└ EDF	EDFNAM	NUMCHN	ESSCAL				
	├ MASS	INTERP	IESSRC:							
			├ CF	CFNAME	IDMAT	ESSCAL				
			├ TF	TFNAME	IDMAT	ESSCAL				
			└ EDF	EDFNAM	NUMCHN	IDMAT	ESSCAL			
	├ TE	INTERP	IESSRC:							
			├ CF	CFNAME	IDMAT	ESSCAL				
			└ TF	TFNAME	IDMAT	ESSCAL				
	└ WE	INTERP	IESSRC:							
			├ CF	CFNAME	ELEV	ISAUTOPT:				
						├ UseCF	CFNAME	IFOGOPT	ESSCAL	
						├ UseCONST	DIAM	IFOGOPT	ESSCAL	
						└ UseSC	IFOGOPT	ESSCAL		
			├ TF	TFNAME	ELEV	ISAUTOPT:				
						├ UseCF	CFNAME	IFOGOPT	ESSCAL	
						├ UseCONST	DIAM	IFOGOPT	ESSCAL	
						└ UseSC	IFOGOPT	ESSCAL		
			└ EDF	EDFNAM	NUMCHN	ELEV	ISAUTOPT:			
							├ UseCF	CFNAME	IFOGOPT	ESSCAL
							├ UseCONST	DIAM	IFOGOPT	ESSCAL
							└ UseSC	IFOGOPT	ESSCAL	

CV_DIF – Gas Diffusion Model

Optional.

This record turns on or off the gas diffusion model for CVs. If on, gaseous species can diffuse between CVs through connecting flowpaths.

(1) IDiffuse

Switch diffusion model on or off.

(a) 0

Off.

(b) 1

On.

(type = integer, default = 0, units = dimensionless)

Example

```
CV_DIF   1   !Turn on gas diffusion model
```

2.2 MELCOR User Input

The CVH database cannot be modified at a MELCOR restart at this time. However, a diagnostic tracing option may be activated as an aid in identifying problem areas in a calculation. Note that there is a MELCOR input – CVH_ATMCS (see MELGEN input for the description to be entered in MELCOR. This option may also be activated from the interactive menu described in the Executive (EXEC) Package Users' Guide. In either case, the user may limit the number of steps for which the trace will be active.

CVH_TRACE – Calculational Trace Activation

Optional.

If this record is present, the progress of the solution of the CVH hydrodynamic equations will be tracked, and informative messages will be written to the diagnostic file (MELDIA). See Section 8.2 for a discussion of these messages. Tracing can also be activated from the interactive menu. In either case, the user has the option of limiting the number of cycles (timesteps) for which it is active. The status of tracing is not recorded in the restart file, and it is, therefore, off unless activated during the current execution. The output is voluminous, and an unrestricted trace should only be activated for short runs.

(1) NTRACE

Number of cycles after which the trace will be terminated.

(type = integer, default = 999999999, units = none)

Example

```
CVH_TRACE    200
```

CVH_ALLOWCOLDATM – Allow dry atmosphere temperature below 273.15

Optional

If this record is present, a calculation will be allowed to continue with atmosphere temperatures below 273.15. It will still fail if there is water in the atmosphere, or if the pool temperature is below 273.15, because the water equation of state is invalid there.

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters of the physics models that would otherwise require modification of the FORTRAN source code. See Section 7 of the MELCOR EXEC Users' Guide for a description of their use.

3.1 CVH Sensitivity Coefficients

The sensitivity coefficients for all hydrodynamic modeling involving the CVH and FL packages have identifier numbers from 4400 to 4599. This section contains brief descriptions of these sensitivity coefficients. Additional information, including references for models and equations, may be found in the Thermal Hydraulic (CVH and FL) Packages Reference Manual. Appendix A in that Reference Manual provides information to aid in locating the specific locations where each sensitivity coefficient is discussed.

The CVH package may use two or more smaller steps to advance through a MELCOR system timestep; each such step is referred to as a subcycle. Timestep controls are based on changes over the system timestep. If an iteration fails to converge or if an error is encountered in the thermodynamics, CVH will attempt to repeat the calculation with a reduced subcycle step before modifying the MELCOR system timestep.

The following record, CVH_SC, is used to change sensitivity coefficients associated with CVH package.

CVH_SC – CVH Sensitivity Coefficients Input Record

Optional.

This record allows users to change the default CVH sensitivity coefficient values to user-input ones. This record is not required.

(1) NUMSTR

Number of data strings.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the CVH sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only; no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
CVH_SC  5  ! N   SCnumber    Value         Index
             1   4407        0.7           11
```

Note: Several sensitivity coefficients in the CVH database are currently unused. Input for unused coefficients is accepted (for compatibility with older versions of MELCOR), but has no effect on the calculation. For completeness, these coefficients are still listed in the discussion below.

4400 – Timestep Control

These coefficients are used to control the system timestep. Array elements 4–11 are used to define limits on the maximum change desired or permitted in the pressure or temperature in any volume during a single timestep. For example, the CVH package will attempt to control the maximum pressure change in any volume to satisfy $\left|P^n - P^o\right| \le C4400(4)\,P + C4400(5)$ by requesting a system timestep for the next timestep, such that this condition would not be violated for the rates of change calculated on the last completed timestep. If, however, the result on completion of a

timestep is that $\left|P^n - P^o\right| > C4400(6)\,P + C4400(7)$ for some volume, CVH will request a MELCOR fallback, and the entire advancement will be repeated with an appropriately reduced timestep. Changes in pool and atmosphere temperatures are treated similarly.

The default values for C4400(4–11) were chosen to permit a safety factor of roughly 2 between the desired maximum changes and the changes which will lead to a fallback. A similar safety factor should be maintained if the default values are modified.

(1) Maximum fraction of material Courant timestep allowed.

(default = 0.5, units = dimensionless, equiv = DTFRCC)

(2) Timestep reduction fraction used when the number of significant digits from the matrix solver is less than two. If the matrix is poorly conditioned, the number of significant digits for the velocities may be so small as to be meaningless. When the number of significant digits is less than two, then the system timestep will be reduced for the next cycle.

(default = 0.9, units = dimensionless, equiv = DTFRCD)

(3) The fraction of the (old) system timestep used when the equation of state package returns an error for a calculation involving no flow paths; the cycle is repeated with the system timestep cut to this fraction of its previous value. (For a calculation involving flow paths, CVH will first reduce the subcycle step.)

(default = 0.15, units = dimensionless, equiv = DTEOSF)

(4) Relative change contribution to the maximum desired pressure change in any volume during a single timestep. Used with C4400(5).

(default = 0.05, units = dimensionless, equiv = DPROK)

(5) Absolute change contribution to the maximum desired pressure change in any volume during a single timestep. Used with C4400(4).

(default = 0.0, units = Pa, equiv = DPAOK)

(6) Relative change contribution to the maximum permitted pressure change in any volume during a single timestep. Used with C4400(7).

(default = 0.1, units = dimensionless, equiv = DPRCUT)

(7) Absolute change contribution to the maximum permitted pressure change in any volume during a single timestep. Used with C4400(6).

(default = 0.0, units = Pa, equiv = DPACUT)

(8) Relative change contribution to the maximum desired temperature change in any volume during a single timestep. Used with C4400(9).

(default = 0.1, units = dimensionless, equiv = DTROK)

(9) Absolute change contribution to the maximum desired temperature change in any volume during a single timestep. Used with C4400(8).

(default = 1.0, units = K, equiv = DTAOK)

(10) Relative change contribution to the maximum permitted temperature change in any volume during a single timestep. Used with C4400(11).

(default = 0.2, units = dimensionless, equiv = DTRCUT)

(11) Absolute change contribution to the maximum permitted temperature change in any volume during a single timestep. Used with C4400(10).

(default = 1.0, units = K, equiv = DTACUT)

4401 – Velocity Convergence Criteria

The velocities are calculated by inverting a system of linear equations. The coefficients in the matrix and the inhomogeneity vector are functions of the velocity. The velocities are iterated until the difference between successive iterations is sufficiently small that $\left|v^{i} - v^{i-1}\right| \le C4401(1)\left(\left|v^{i}\right| + \left|v^{i-1}\right|\right) + C4401(2)$ for all flows (the superscript refers to the iteration number). Additional tests identify reversed flows.

(1) Relative convergence criterion.

(default = .09, units = dimensionless, equiv = CVELR)

(2) Absolute velocity convergence criterion.

(default = 0., units = m/s, equiv = CVELA)

(3) Maximum number of iterations permitted before solution is repeated with a decreased (subcycle) timestep. The default value of 0.0 is used to specify a limit scaled to the size of the problem; the actual value used is 8.0 + the number of flow paths in the nodalization.

(default = 0., units = dimensionless, equiv = XPASMX)

(4) Number of iterations after which velocities will be considered converged if there is no significant effect (less than 0.05 %) on pressures. A value of 0.0 gives the default treatment, which is not to consider the relaxed convergence criterion. (Treatment in versions prior to 1.8.2 corresponded to a value of 8.0.)

(default = 0.0, units = dimensionless, equiv = XPASP)

4402 – Minimum Velocity to be Considered for Choking Calculation

Velocities calculated from the flow (momentum) equations subjected to limits imposed by critical flow correlations. The tests will be bypassed unless either the pool velocity or the atmosphere velocity is greater than a threshold value.

(1) Velocity below which choking tests are bypassed.
(default = 20., units = m/s, equiv = CVELMN)

4404 – Friction Factor Parameters

A two-phase friction factor is calculated for each flow path segment. Laminar flow is assumed if the Reynolds number is less than C4404(14), turbulent flow is assumed if the Reynolds number is greater than C4404(15), and a transition region is assumed for Reynolds numbers between these values. The friction factor for laminar flow is determined by dividing the value of the variable SLAM by the value of the Reynolds number. The default value of SLAM is C4404(13). The Colebrook-White equation

$$\frac{1}{\sqrt{f}} = C4404(1) - C4404(2)\log_{10}\left(C4404(3)\frac{e}{D} + \frac{C4404(4)}{\mathrm{Re}\sqrt{f}}\right)$$

is used to define the friction factor, *f*, for turbulent flow, where *e* is the surface roughness and *D* is the hydraulic diameter.

Re is the Reynolds number based on the hydraulic diameter and a two-phase mixture viscosity, μ_M, defined by

$$\mu_M = \mu_P\left[C4404(6) - \alpha\right]\left[C4404(10) + C4404(11)\ \alpha\right] + \alpha\ \mu_A$$

where μ_P is the pool viscosity, α is the void fraction, and μ_A is the atmosphere viscosity. Note that, if C4404(6) = 1.0, it has the proper limits for $\alpha \to 0.0$ and $\alpha \to 1.0$.

(1) Coefficient in Colebrook-White correlation.
(default = 3.48, units = dimensionless, equiv = none)

(2) Coefficient in Colebrook-White correlation.
(default = 4.0, units = dimensionless, equiv = none)

(3) Coefficient in Colebrook-White correlation.
(default = 2.0, units = dimensionless, equiv = none)

(4) Coefficient in Colebrook-White correlation.
(default = 9.35, units = dimensionless, equiv = none)

(5) Value used in iterative solution. <u>Should not be modified</u>.
(default = 1.0/ln(10) = 0.43429448190325182765, units = dimensionless, equiv = none)

(6) Coefficient in two-phase viscosity correlation. Must be 1.0 to get proper limit for $\alpha = 1.0$.
(default = 1.0, units = dimensionless, equiv = none)

(7) Value used in iterative solution. Should not be modified.
(default = 14.14, units = dimensionless, equiv = XINOLD)

(8) Value used in iterative solution. Should not be modified.
(default = 0.0005, units = dimensionless, equiv = RELC)

(9) Value used in iterative solution. Should not be modified.
(default = 0.0, units = dimensionless, equiv = RELA)

(10) Coefficient in two-phase viscosity equation.
(default = 1.0, units = dimensionless, equiv = none)

(11) Coefficient in two-phase viscosity equation.
(default = 2.5, units = dimensionless, equiv = none)

(12) If the void fraction is less than this number, the atmosphere friction factor is set to zero. For greater void fractions, the atmosphere friction factor is linearly interpolated between zero and the value given by the basic model.
(default = 0.9, units = dimensionless, equiv = none)

(13) Default coefficient for calculating laminar flow friction factor.
(default = 16.0, units = dimensionless, equiv = none)

(14) Maximum value of Reynolds number for laminar flow.
(default = 2000.0, units = dimensionless, equiv = none)

(15) Minimum value of Reynolds number for turbulent flow.
(default = 5000.0, units = dimensionless, equiv = none)

4405 – SPARC Bubble Physics Parameters

The user may specify on an FLnnn02 input record that mass and energy exchange terms should be calculated for hot gases entering a pool from a flow path which connects to a control volume below the pool surface in that volume. Such gases are not considered to reside in the pool, but their interactions will be accounted for before the exiting gases are allowed to join the atmosphere in the control volume. Parametric models are used in MELCOR; efficiency terms are calculated to account for bubble rise distance and liquid subcooling. The bubble rise efficiency is given by

$$\varepsilon_z = \max\left\{0.0, \min\left[1.0, \frac{\Delta z - C4405(1)}{hC4405(2)}\right]\right\}$$

where z is the distance that bubbles rise through the pool and h is the flow path opening height. The water temperature efficiency is given by

$$\varepsilon_T = \max\left\{0.0, \min\left[1.0, \frac{T_{sat}(P) - T_P - C4405(3)}{C4405(4)}\right]\right\}$$

where $T_{sat}(P)$ is the saturation temperature at total pressure and T_P is the pool temperature. The net efficiency used is the product of ε_z and ε_T.

(1) Zero efficiency bubble rise height.
(default = 0.01, units = m, equiv = DZZERO)

(2) Full efficiency bubble rise height in terms of flow path junction heights.
(default = 1.0, units = dimensionless, equiv = DZEFF)

(3) Zero efficiency temperature difference between the pool and saturation temperature at the junction elevation.
(default = 0.1, units = K, equiv = DTMPZR)

(4) Full efficiency temperature difference between the pool and saturation temperature at the junction elevation.
(default = 5.0, units = K, equiv = DTEFF)

(5) Relative humidity of the steam at the surface.
(default = 0.99, units = dimensionless, equiv = PEFF)

4406 – Maximum Allowed Fog Density

The maximum mass of liquid water in the atmosphere (fog) divided by the volume of the atmosphere that will not be permitted to be larger than FOGDEN. Any excess liquid water is moved to the pool.

(1) Maximum fog density.
(default = 0.1, units = kg/m^3, equiv = FOGDEN)

4407 – Pool/Atmosphere Heat/Mass Transfer Parameters

Energy and mass transfer occurs between the pool and atmosphere of a nonequilibrium volume when their temperatures differ. The models used are outlined briefly in Section 1.3 of this report and described in greater detail in the CVH/FL Reference Manual.

The heat transfer coefficient for each phase is given by

$$h = \max(h_{forced}, h_{free}, k/l)$$

Here h_{forced} is given by the TRAC correlation for horizontal stratified flow,

$$h_{forced} = C4407(2)\rho_{\varphi} c_{p,\varphi} v_{V,\varphi}$$

and h_{free} is derived from the correlations for natural (free) convection for horizontal surfaces

$$h_{free,A} = \max\left[C4407(3)(Gr\ \mathrm{Pr})^{C4407(4)}, C4407(5)(Gr\ \mathrm{Pr})^{C4407(6)}\right]\frac{k_A}{X_A}$$

$$h_{free,P} = \max\left[C4407(7)(Gr\ \mathrm{Pr})^{C4407(8)}, C4407(9)(Gr\ \mathrm{Pr})^{C4407(10)}\right]\frac{k_P}{X_P}$$

where ρ is the density, c_{p} is the specific heat, v_{V} is the volume velocity, k is the thermal conductivity, X is the characteristic length, φ refers to the phase (pool [P] or atmosphere [A]), and Gr and Pr are the Grashof and Prandtl numbers, respectively.

The term k/l represents the conduction limit.

(1) Bubble rise velocity.
(default = 0.3, units = m/s, equiv = VELBUB)

(2) Coefficient for TRAC heat transfer coefficient correlations.
(default = 0.02, units = dimensionless, equiv = TRACC)

(3) Coefficient for turbulent atmosphere heat transfer correlations.
(default = 0.14, units = dimensionless, equiv = CTLATA)

(4) Exponent for turbulent atmosphere heat transfer correlations.
(default = 1/3, units = dimensionless, equiv = ETLATA)

(5) Coefficient for laminar atmosphere heat transfer correlations.
(default = 0.54, units = dimensionless, equiv = CTLALA)

(6) Exponent for laminar atmosphere heat transfer correlations.
(default = 1/4, units = dimensionless, equiv = ETLALA)

(7) Coefficient for turbulent pool heat transfer correlations.
(default = 0.27, units = dimensionless, equiv = CTLATP)

(8) Exponent for turbulent pool heat transfer correlations.
(default = 1/4, units = dimensionless, equiv = ETLATP)

(9) Coefficient for laminar pool heat transfer correlations.
(default = 0.27, units = dimensionless, equiv = CTLALP)

(10) Exponent for laminar pool heat transfer correlations.
(default = 1/4, units = dimensionless, equiv = ETLALP)

(11) Maximum void fraction permitted in a control volume pool (enforcement is approximate).

(default = 0.4, units = dimensionless, equiv = ALPMAX)

(12) Maximum fraction of atmospheric water vapor permitted to condense during a (subcycle) timestep.

(default = 0.9, units = dimensionless, equiv = CFRMAX)

(13) Upper limit to be imposed on surface-to-volume ratio of small pools in calculation of bubble separation.

(default = 1000.0, units = dimensionless, equiv = SVRMAX)

4408 – Pressure Iteration Parameters

Several submodels in the flow solution may be turned off if the user so desires. This capability is intended primarily for use in debugging by the code developers. The first element of the sensitivity coefficient array, C4408(1), entered as a single real number, is interpreted as six packed decimal digits, each controlling a single model. Setting any digit $\neq 0$ disables the corresponding model. Any or all of the digits may be nonzero as desired.

Setting the 1,000,000s digit $\neq 0$ disables the revised treatment of non-equilibrium thermodynamics that prevents the collapse of a boiled-up pool in a volume with a very small atmosphere and the possible revaporization of a very small pool. Here "very small" is defined by VPHEQL, element 5 of sensitivity coefficient array 4411. The resulting behavior corresponds to the model (a switch to equilibrium thermodynamics) used in previous MELCOR versions through early releases of 1.8.6.

Setting the 100,000's digit $\neq 0$ disables the implicit treatment of bubble separation from two-phase pools. The resulting explicit calculation is similar (but not identical) to the model used in previous MELCOR versions through 1.8.2.

Setting the 10,000's digit $\neq 0$ disables the limitation on void fraction by depletion of the atmosphere.

Setting the 1,000's digit $\neq 0$ disables the redefinition of old velocities to preserve the old volume flow as the void fraction changes during the iteration.

Setting the 100's digit $\neq 0$ disables the calculation of momentum exchange between the pool and the atmosphere sharing a flow path.

Setting the 10's digit $\neq 0$ results in using pressures at the nominal junction elevation in the acceleration term of the momentum equation, rather than at the average elevation of a phase in the junction opening.

Setting the 1's digit $\neq 0$ eliminates calculation of the implicit contributions to the head terms of the momentum equation.

(1) Submodel flags, controlling submodels in the flow solution.

(default = 0., units = dimensionless, equiv = XOPTN)

(2) A parameter used as a convergence criterion for two-phase states in the pressure iteration. It is also used to allow the subcycle timestep to be increased whenever the convergence of pressures is much tighter than required.

(default = 0.005, units = dimensionless, equiv = XXCONV)

4409 – Limits and Tolerances for Time-Specified Volumes

Certain limits are placed on values input from functions or tables to define the properties of time-specified volumes. If a pool is present, it must also be between the freezing point and the critical point. In addition, discrepancies greater than round-off are tolerated in the consistency of input to allow for such things as imprecise interpolation in coarse tables. Of particular concern is the need to be able to specify a saturated state when both temperature and pressure are input. Values outside these limits and/or tolerances will result in failure to create a restart file in MELGEN or in termination of the calculation in MELCOR.

(1) Minimum temperature.

(default = 1.0, units = K, equiv = TMIN)

(2) Maximum temperature.

(default = 1.0E4, units = K, equiv = TMAX)

(3) Minimum volume pressure.

(default = 1.0, units = Pa, equiv = PMIN)

(4) Maximum volume pressure.

(default = 1.0E8, units = Pa, equiv = PMAX)

(5) Tolerance on the sum of NCG mole fractions compared to 1.0.

(default = 0.001, units = dimensionless, equiv = TOLMFR)

(6) Tolerance on the saturation temperature, within which input corresponding to a superheated pool or a subcooled atmosphere will be treated as saturated.

(default = 1.0, units = K, equiv = TOLTS)

4410 – Vapor Velocity Enhancement during Direct Containment Heating

The largely parametric treatment of High Pressure Melt Ejection (HPME) in the FDI package does not model coupling between the velocity of airborne debris and that of the volume atmosphere. One potentially important effect of the neglected coupling would be to increase the local gas velocities and, therefore, to enhance heat transfer from the atmosphere to structures in the volume. These coefficients (used only in volumes where HPME is occurring) allow a parametric treatment of the enhancement

through modification of the velocities that will be used in evaluating heat transfer coefficients.

(1) Enhancement factor applied to the volume-averaged vapor velocities during direct containment heating.

(default = 1.0, units = dimensionless, equiv = VMULT)

(2) Minimum temperature of airborne debris for application of the enhancement.

(default = 1500., units = K, equiv = TDBMIN)

4411 – Limits and Tolerances for Iterations in the CVT Package

These coefficients are used to control the iterative calculations in evaluation of the mixed-material equation of state in the CVT package.

(1) Tolerance (relative) in determination of the temperature, given the density and the specific internal energy.

(default = 1.0E-6, units = dimensionless, equiv = TOLTR)

(2) Tolerance (relative) in matching the pressures of the pool and the atmosphere in a control volume. (The volume pressure is returned as a compressibility-weighted average of the two.)

(default = 2.0E-5, units = dimensionless, equiv = TOLPR)

(3) Tolerance (absolute) in matching the pressure of a subcooled pool to that of the atmosphere in a control volume. The alternate tolerance is made necessary by the near incompressibility of liquid water. The default value is slightly greater than the pressure change associated with round-off in liquid density at atmospheric pressure on a 32-bit computer and represents the limit in precision with which the pressure of a subcooled liquid can be determined from its density. Because the volume pressure is returned as a compressibility-weighted average of pool and atmosphere pressures, the effect on the final pressure is greatly reduced if the volume contains a significant volume of atmosphere. This tolerance is also considered in the control of timesteps based on the discrepancy between the predicted and the final volume pressures.

(default = 300.0, units = Pa, equiv = TOLPA)

(4) Minimum volume fraction to be assigned to the pool or the atmosphere in a control volume after thermodynamic calculations have been completed. Although MELCOR 1.8.2 used a (hard-wired) value of 1.0E-6, a nonzero value will result in inconsistent properties for a very small pool or

atmosphere and interfere with convergence of iterations. This coefficient is provided to allow return to the older model, should the need arise.

(default = 0.0, units = dimensionless, equiv = VPHMIN)

(5) Minimum (estimated) volume fraction of the pool or the atmosphere below which equilibrium thermodynamics will be enforced, regardless of the option specified by ICVTHR on the CV_THR input record.

(default = 1.0E-6, units = dimensionless, equiv = VPHEQL)

4412 – Limits and Tolerances for Iterations in the CVH Package

These coefficients are used to control the iterative calculations in solution of the hydrodynamic equations in the CVH package.

(1) Permitted discrepancy between the implicitly projected value and the final value of the pool void fraction in any volume during solution of the implicit flow equations.

(default = 0.01, units = dimensionless, equiv = ALPTOL)

4413 – Flow Blockage Friction Parameters

These parameters are used to calculate the friction loss in a flow path that has been at least partially blocked by debris. The pressure drop will be based on a generalized Ergun equation in the form

$$K_{eff} = \left[C4413(1) + C4413(2)\left(\frac{1-\varepsilon}{\mathrm{Re}}\right) + C4413(3)\left(\frac{1-\varepsilon}{\mathrm{Re}}\right)^{C4413(4)} \right] \frac{(1-\varepsilon)L}{\varepsilon D}$$

where ε is the porosity and D is the particle diameter, and $\mathrm{Re} = \rho j D / \mu$ is the Reynolds number based on the superficial velocity (volumetric flux) j, the fluid viscosity μ, and the particle diameter.

(1) Coefficient of the turbulent term in the generalized Ergun equation.

(default = 3.5, units = dimensionless, equiv = CTERG)

(2) Coefficient of the laminar term in the generalized Ergun equation.

(default = 300.0, units = dimensionless, equiv = CLERG)

(3) Coefficient of the Achenbach term in the generalized Ergun equation.

(default = 0.0, units = dimensionless, equiv = CCACH)

(4) Exponent in the Achenbach term in the generalized Ergun equation.
(default = 0.4, units = dimensionless, equiv = CPACH)

(5) Minimum porosity to be used in evaluating the correlation, imposed as a bound before K_{eff} is evaluated.
(default = 0.05, units = dimensionless, equiv = PORMIN)

4414 – Minimum Hydrodynamic Volume Fraction

This parameter defines a fraction of the initial hydrodynamic volume in each segment of the volume/altitude table of a control volume (as specified on the CVHnnnBk records) that will be considered as available to hydrodynamic materials. This volume is preserved, regardless of virtual volume changes resulting from relocation of nonhydrodynamic materials such as core debris.

(1) Minimum fraction of the initial volume in each segment of the volume/altitude table of a control volume that will always be available to hydrodynamic materials.
(default = 0.01, units = dimensionless, equiv = FVMIN)

4415 – Criteria for Solving the Flow Equations in Sparse Form

These coefficients are used to specify the conditions for use of the sparse equation solver and to define the associated convergence tolerances.

(1) The maximum fraction of nonzero coefficients for use of the sparse form. A value of 0.0 assures that the direct solution will always be used, while a value of 1.0 assures that the iterative solution will always be used.
(default = 1.0, units = dimensionless, equiv = FRMAXS)

(2) Convergence tolerance for the iterative solver.
(default = 1.0E-7, units = dimensionless, equiv = TOLSLP)

(3) Maximum number of iterations permitted for the iterative solver.
(default = 500.0, units = dimensionless, equiv = XITMAX)

(4) Convergence option for the iterative solver.
= 0.0 : iteration stops when the 2-norm of the residual divided by the 2-norm of the right-hand side is less than TOLSLP.
$\neq$ 0.0 : iteration stops when the 2-norm of M_{inv} times the residual divided by the 2-norm of M_{inv} times the right-hand

side is less than TOLSLP, where M_{inv} is the inverse of the diagonal of the coefficient matrix.

(default = 0.0, units = dimensionless, equiv = TOLTYP)

4416 – Parameters limiting over-extraction of pool or atmosphere.

These coefficients contains parameters limiting over-extraction of pool or atmospheres when moving volumes through flow paths.

(1) Allow over-extraction by no more than a fraction DFPMIN of the total volume without cutting step

(default = 1.0e-6, units = dimensionless, equiv = DFPMIN)

(2) Allow overextraction by no more than a fraction DFAMIN of the total volme without cutting step

(default = 1.0e-6, units = dimensionless, equiv = DFAMIN)

(3) To be used by developer only

(default = 0.0, units = dimensionless, equiv = DFRMIN)

(4) To be used by developer only

(default 0.0 =, units = dimensionless, equiv = AFRMIN)

4417 –Data for controling the Umfpack solver

These coefficients are used to control the Umfpack solver

(1) The number of iterative refinement steps to take following the direct solve.

(default = 2, units = dimensionless, equiv = NIRSTPS)

(2) Parallel solver option

0 means solve serially, 1 means use tasks, 2 means use thread

(default =0, units = dimensionless, equiv = UPPARTYPE)

4418 –Data for controlling the additive Schwarz solver

These coefficients are used to control the additive Schwaz solver

(1) The maximum number of iterations.

(default = 100, units = dimensionless, equiv =ASNMAXIT)

(2) Jacobi block size. The maximum number of rows in each solver block

(default = 20, units = dimensionless, equiv = JBLKSIZ)

(3) Parallel solver option

0 means solve serially, 1 means use tasks, 2 means use threads.

(default = 0, units = dimensionless, equiv = ASPARTYPE)

4419 – Data for controling biconjugate gradient solver

These coefficients are used to.

(1) Maximum number of iterations in solver

(default = 100, units = dimensionless, equiv =BGNMAXIT)

(2) Parallel solver option

0 means solve serially, 1 means use tasks, 2 means use threads

(default = 0, units = dimensionless, equiv = BGPARTYPE)

4420 – Flag for parallelization of the cvh package

These coefficients are used to.

(1) The

(default =, units = dimensionless, equiv =)

4421 – Data for 2-sided quench model

These coefficients are used to improve stability of the 2-sided quench model. Since heat transfer between surfaces is explicit, this sensitivity coefficient is needed.

(1) Maximum temperature change on a time step. If temperature change exceeds this value, the COR package will drop the time step.

(default = 50.0, units = dimensionless, equiv =NIRSTPS)

(2) Maximum change in heat flux from opposite side calculated for a time step. If change in heat flux exceeds this value, the COR package will drop the time step.

(default = 10.0, units = dimensionless, equiv =UPPARTYPE)

4422 – Flag for sorting flow paths by Flow Path number

This flag determines how and if the flow paths are to be sorted. Note that the order of flow paths can lead to differences in results. Sorting them by number tends to yield consistent results as new flow paths are added to the model since position in the input deck is no longer important. The user can also choose to randomly shuffle the position of flow paths in the database for testing purposes.

(1) Sorting flag,
0.0 no sort, 1.0 sort by flow path number, 2.0 randomly sort flow paths.
(default = 1.0, units = dimensionless, equiv = `FPSort`)

(2) Random number seed used in generating a random yet reproducible sorting order. A value of 0.0 indicates that MELCOR will generate a random number seed based on the system clock time.
(default = 0.0, units = dimensionless, equiv = `none`)

4450 –Parameters to choose between old and new procedure to calculate critical flow

This coefficient is used to.enable the IBRAE model for calculating critical flow

(1) A value of 1 enables IBRAE model
(default = 0, units = dimensionless, equiv = `CHKFLAG`)

4500 – Parameters in the Flashing Model for Sources and Flows

The partition of liquid water between the pool and fog will be calculated from a Rosin-Rammler distribution

$$\frac{M(d > d_P)}{M_{Total}} = \exp\left\{-\left[\Gamma\left(\frac{k-1}{k}\right)\right]^{-k}\left(\frac{d_p}{\overline{d}_S}\right)^k\right\}$$

Here $\overline{d}_S$ is the Sauter mean droplet diameter, the single diameter that would match the surface-to-volume ratio for the distribution, k is an exponent, and Γ is the gamma function. If the RN1 package is active, the same distribution will be used to distribute the fog over size bins (sections) of water-class aerosols. The default value of k is that suggested by Eltkobt.

(1) Efficiency of flashing used as a multiplier on the transformation.
(default = 1.0., units = none, equiv = FlashEff)

(2) Fog fraction. If > 0.0, it will be used to override the value calculated from the Rosin-Rammler distribution.

(default = -1.0., units = none, equiv = FogFract)

(3) Sauter mean droplet diameter.
(default = 65.0E-6., units = m, equiv = SauterMean)

(4) Maximum diameter for fog if the RN1 package is inactive.
(default = 50.0E-6., units = m, equiv = DiamMax)

(5) Exponent in Rosin-Rammler distribution.
(default = 5.32., units = m, equiv = DiamMax)

4 Plot Variables

The variables in the CVH package that may be used for plot variables are listed and described below. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

CVH-ATM-FR.n	Atmosphere (non-pool) volume fraction in control volume index n. (units = dimensionless)
CVH-CLIQLEV.n	Collapsed liquid elevation in control volume index n. (units = m)
CVH-CPUT	Total central processing unit (CPU) usage by the RUN (time-advancement) portion of the CVH package. (units = s)
CVH-CPUE	CPU usage for edit in the RUN portion of the CVH package. (units = s)
CVH-CPUC	CPU usage for calculations in the RUN portion of the CVH package. (units = s)
CVH-CPUR	CPU usage to process the restart file in the RUN portion of the CVH package. (units = s)
CVH-E.n	Specific internal energy in control volume index n. (units = J/kg)

CVH-E.m.n	Specific internal energy of material m in control volume index n. (units = J/kg)
CVH-ECV.n	Total internal energy in control volume index n. (units = J)
CVH-ECV.m.n	Total internal energy of material m in control volume index n. (units = J)
CVH-H.n	Specific enthalpy in control volume index n. (units = J/kg)
CVH-H.m.n	Specific enthalpy of material m in control volume index n. (units = J/kg)
CVH-LIQLEV.n	Swollen liquid elevation in control volume index n. (units = m)
CVH-MASS.n	Mass of material in control volume index n. (units = kg)
CVH-MASS.m.n	Mass of material m in control volume index n. (units = kg)
CVH-P.n	Pressure of control volume index n. (units = Pa)
CVH-PPART.m.n	Partial pressure of material m in control volume index n. (units = Pa)
CVH-PSAT(TLIQ).n	Saturation pressure of water at the pool temperature in control volume index n. (units = Pa)
CVH-PSAT(TVAP).n	Saturation pressure of water at the atmosphere temperature in control volume index n. (units = Pa)
CVH-QUALITY.n	Quality (non-pool, non-fog mass fraction) in control volume index n.

	(units = dimensionless)
CVH-RHO.n	Net mass density of all materials in control volume index n. (units = kg/m^3)
CVH-RHO.m.n	Mass density of material m in control volume index n. (units = kg/m^3)
CVH-TLIQ.n	Pool temperature in control volume index n. (units = K)
CVH-TOT-E.m	Total internal energy of material m in the CVH package. (units = J)
CVH-TOT-M.m	Total mass of material m in the CVH package. (units = kg)
CVH-TSAT(P).n	Water saturation temperature at the pressure of control volume index n. (units = K)
CVH-TSAT(A).n	Water saturation temperature at the partial pressure of water vapor in the atmosphere of control volume index n. (units = K)
CVH-TVAP.n	Atmosphere temperature in control volume index n. (units = K)
CVH-VELLIQCV.n	Velocity of the pool in control volume index n. (units = m/s)
CVH-VELVAPCV.n	Velocity of the atmosphere in control volume index n. (units = m/s)
CVH-VIRVOLIQ.n	Virtual volume in the pool in control volume index n. (units = m^3)
CVH-VIRVOVAP.n	Virtual volume in the atmosphere in control volume index n.

	(units = m^3)
CVH-VOID-P.n	Void (bubble) volume fraction in the pool of control volume index n. (units = dimensionless)
CVH-VOID-T.n	Total void (non-liquid) volume fraction in control volume index n, including the atmosphere and bubbles in the pool. (units = dimensionless)
CVH-VOLFOG.n	Volume of fog in control volume index n. (units = m^3)
CVH-VOLLIQ.n	Swollen volume of the pool (including any vapor bubbles) in control volume index n. (units = m^3)
CVH-VOLVAP.n	Volume of gaseous atmosphere in control volume index n. (units = m^3)
CVH-X.m.n	Mole fraction of gaseous material m (m ≥ 3) in the atmosphere of control volume index n. (units = dimensionless)

5 Control Function Arguments

The variables in the CVH package that may be used for control function arguments are listed and described below. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

The choices permitted for NameMat always include 'POOL', 'FOG', 'H2O-VAP', or those other materials identified by input to the NonCondensible Gas (NCG) package. In certain cases (see below) the keywords 'TOTAL' (or 'ALL'), 'WATER' are also interpreted to mean, respectively, the total contribution from all materials, or the total contribution from all water phases ('POOL', 'FOG', 'H2O-VAP').

CVH-ATM-FR(NameCV)	Atmosphere (non-pool) volume fraction in control volume NameCV. (units = dimensionless)
CVH-CLIQLEV(NameCV)	Collapsed liquid elevation in control volume NameCV.

	(units = m)
CVH-CPUT	Total CPU usage by the RUN (time-advancement) portion of the CVH package. (units = s)
CVH-CPUE	CPU usage for edit in the RUN portion of the CVH package. (units = s)
CVH-CPUC	CPU usage for calculations in the RUN portion of the CVH package. (units = s)
CVH-CPUR	CPU usage to process the restart file in the RUN portion of the CVH package. (units = s)
CVH-E(NameCV,NameMat)	Specific internal energy of material NameMat in control volume NameCV or specific internal energy in control volume Name if NameMat = TOTAL. (units = J/kg)
CVH-ECV(NameCV,NameMat)	Total internal energy of material NameMat in control volume NameCV or total internal energy in control volume if NameMat = TOTAL. (units = J)
CVH-H(NameCV,NameMat)	Specific enthalpy of material NameMat in control volume NameCV or specific enthalpy in control volume if NameMat = TOTAL. (units = J/kg)
CVH-LIQLEV(NameCV)	Swollen liquid elevation in control volume NameCV. (units = m)
CVH-MASS(NameCV, NameMat)	Mass of material NameMat in control volume NameCV or Mass of material in control volume if NameMat = ALL. (units = kg)
CVH-P(NameCV)	Pressure of control volume NameCV. (units = Pa)

CVH-PPART(NameCV,NameMat)	Partial pressure of material NameMat in control volume NameCV. (units = Pa)
CVH-PSAT(NameCV,key)	Saturation pressure of water at the atmosphere (key = ATM) or at the pool (key = POOL) temperature in control volume NameCV. (units = Pa)
CVH-QUALITY(NameCV)	Quality (non-pool, non-fog mass fraction) in control volume NameCV. (units = dimensionless)
CVH-RHO(NameCV,key)	Mass density of materials in control volume NameCV. The choices allowed for key are ALL - density of all materials ATM - density of atmosphere POOL - density of pool NameMAT - density of material. (units = kg/m^3)
CVH-TLIQ(NameCV)	Pool temperature in control volume NameCV. (units = K)
CVH-TOT-E(NameMat)	Total internal energy of material NameMAT in the CVH package. (units = J)
CVH-TOT-M(NameMat)	Total mass of material NameMat in the CVH package. (units = kg)
CVH-TSAT(NameCV,key)	Water saturation temperature at the pressure (key = P) or Water saturation temperature at the partial pressure of water vapor in the atmosphere (key = A) of control volume NameCV. (units = K)
CVH-TVAP(NameCV)	Atmosphere temperature in control volume NameCV. (units = K)
CVH-VEL(NameCV,key)	Velocity of the atmosphere (key = ATM) or the pool (key = POOL) in control volume NameCV. (units = m/s)

CVH-VIRVOL(NameCV,key)	Virtual volume in the atmosphere (key = ATM) or the pool (key = POOL) in control volume NameCV. (units = m^3)
CVH-VOID(NameCV,key)	Void (bubble) volume fraction in the pool (key = POOL) of control volume NameCV or total void (non-liquid) volume fraction, including the atmosphere (key = ATM) and bubbles in the pool. (units = dimensionless)
CVH-VOLFOG(NameCV)	Volume of fog in control volume NameCV. (units = m^3)
CVH-VOLLIQ(NameCV)	Swollen volume of the pool (including any vapor bubbles) in control volume NameCV. (units = m^3)
CVH-VOLVAP(NameCV)	Volume of gaseous atmosphere in control volume NameCV. (units = m^3)
CVH-X(NameCV, NameMat)	Mole fraction of gaseous material NameMat in the atmosphere of control volume NameCV. (units = dimensionless)

6 Example Input

This section gives several examples of input to the CVH package. Anything following an exclamation point is a comment.

The first example is for a control volume with user-assigned name "VOLUME ONE." It uses nonequilibrium thermodynamics, specifies horizontal flow, and is part of the reactor coolant system. (The flow direction is not used by any model; designation as part of the reactor coolant system is used, if at all, in organizing RN package output.) The volume contains 150 m^3, is 10 m high with a base at 0 m, and has a constant horizontal cross-section. Its flow area for the purposes of defining an average velocity is 12.5 m^2. It is initially filled to the 9.5 m elevation with saturated liquid water at 7.0 MPa, and the remaining space is filled by saturated water vapor.

```
CV_ID   'VOLUME ONE' 'RCS'
CV_ARE  NOCF 12.5
CV_THR  NONEQUIL FOG ACTIVE
CV_PAS  SEPARATE POOLANDATM SATURATED SATURATED
CV_PTD  PVOL 7.0E6
CV_BND  ZPOL 9.5
```

```
CV_VAT  2 !N  CVZ   CVVOL
           1  0.0   0.0    ! Bottom at 0 m
           2  10.0  150.0  ! 150 m3 total volume
```

If user wants to define a SUPERHEATED atmosphere, the CV_PAS record should be changed to the following form,

```
CV_PAS  SEPARATE POOLANDATM SATURATED SUPERHEATED
```

and the following record is also required in this case.

```
CV_AAD   TATM   650.0      ! Superheated
```

The atmosphere will consist of superheated water vapor at 650 K and 7.0 MPa.

Conditions in the volume can be made time-independent by replacing the former CV_THR record with

```
CV_THR  NONEQUIL FOG TIME-INDEP
```

A more complicated shape, with only a third of the total volume in the lower half of the volume height, can be defined if record CV_VAT is replaced by

```
CV_VAT  2 !N  CVZ   CVVOL
           1  5.0   50.0   ! 50 m3 below 5 m
           2  10.0  150.0
```

The next example defines a volume with user-assigned name "WET WELL". It uses nonequilibrium thermodynamics. Its volume is 8000 m^3, with a constant cross-section between elevations of -25 m and -5 m. The initial pressure is 110 kPa, and the volume is filled to the -15 m elevation with subcooled water at 313 K. The atmosphere, at 323 K, has a relative humidity of 90 percent. The balance consists of noncondensible gases, materials N2 and H2; the mole fractions are 80 percent and 20 percent, respectively.

```
CV_ID   'Wet Well'
CV_THR  NONEQUIL FOG ACTIVE
CV_PAS  SEPARATE POOLANDATM SABCOOLED SUPERHEATED
CV_PTD  PVOL 1.1E5   ! Pressure
CV_PAD  313.0        ! Pool temp
CV_AAD  TATM 323.0   ! Atmos temp
CV_BND  ZPOL -15.0   ! Pool surface
CV_VAT  2 ! N  CVZ   CVVOL
            1  -25.0 0.0
            2  -5.0  8.0E3
CV_NCG  2  RHUM  0.9 ! RHUM implies PH2O
        1  N2  0.8
        2  O2  0.2
```

The partial pressure of water vapor in the atmosphere will be 0.9 of the saturation pressure (12.2 kPa) at 323 K, or 11.0 kPa. Noncondensible gases will make up the remaining 99.0 kPa of the total pressure, with partial pressures in a 4-to-1 ratio. Note that these gases must be defined by input to the NCG package. For example, the records

```
NCG_ID  N2
```

```
NCG_ID  O2
NCG_ID  H2
```

will define material 4 as N_2, material 5 as O_2, and material 6 as H_2.

To add a H_2 source to the volume "Wet Well", at a constant rate of 1 kg/s and a temperature of 800 K, simply add source records and tabular function definitions, such as

```
CV_SOU 2 ! N, SourceInfo  1 MASS RATE TF 'H2-RATE'   H2  1.0
         2 TE   RATE TF 'H2-TEMP'   H2  1.0
...
!
TF_INPUT
TF_ID   'H2-RATE' 1.0
TF_TAB  1 ! N  X     Y
            1 0.0   1.0 ! Constant value 1.0
!
TF_ID   'H2-TEMP' 1.0
TF_TAB  1 ! N  X     Y  ! Constant value 800.0
            1 0.0   800.0
```

See the TF Package Users' Guide for information on more complicated tables. Control functions could also be used. An example will be shown later.

The next example defines a time-specified volume with user-assigned name "BOUNDARY". It has a volume of 10 m^3, between elevations of 1 m and 2 m, and is assigned to the class "miscellaneous 1" for RN accounting purposes. It contains a pool of subcooled water with pressure and temperature given by channels 3 and 5 of external data file, "EDF7", respectively. The surface elevation as a function of time is given by the tabular function TF12. The atmosphere temperature is given by channel 4 of the same external data file. It is saturated (or superheated); the composition of any noncondensible gases present is defined by control functions CF3 and CF8 (perhaps to match the composition in some other control volume).

```
CV_ID   BOUNDARY 'miscellaneous 1'
CV_THR  NONEQUIL FOG PROP-SPECIFIED
CV_PTD  PVOL EDF EDF7 3
CV_PAD  EDF  EDF7 5
CV_AAD  TATM EDF7 4
CV_BND  ZPOL TF TF12
CV_VAT  2 !N  CVZ   CVVOL
           1  1.0   0.0
           2  2.0   10.0
CV_NCG 2 SAT !N  GAS NAME KEY NAME
            1      N2    CF  CF3
            2      O2    CF  CF8
```

The various tabular and control functions and the external data file must, of course, be defined. See the TF, CF, and EDF Package Users' Guides for discussion of the required input.

A further example defines a source of subcooled water to the volume "VOLUME ONE" with properties corresponding to the water in the volume "BOUNDARY"—that is, with the pressure and temperature given by channels 3 and 5 of external data file "EDF7",

respectively. The cumulative mass to be added is given as a function of time by control function "INT-MASS-SOURCE".

```
CV_SOU 2 !N, SourceInfo
          1 MASS INTEGRAL CF 'INT-MASS-SOURCE POOL'
          2 PE   RATE     CF 'ENERGY-RATE'
...
!
CF_INPUT
CF_ID   'INT-MASS-SOURCE' . . . ! Input as appropriate
...
CF_ID   'POOL-RATE'  DER-F
CF_SAI  1.0
CF_ARG  1 ! NARG CHARG       ARSCAL  ARADCN
CF-VALU('INT-MASS-SOURCE')   1.0     0.0
!
CF_ID   'ENERGY-RATE'  MULTIPLY
CF_SAI  1.0
CF_ARG  2 ! NARG      CHARG          ARSCAL  ARADCN
            1 CF-VALU('POOL-RATE')     1.0     0.0
2 CVH-H(BOUNDARY,POOL)     1.0     0.0 ! pool in CV
```

This input calculates the appropriate rate of the enthalpy addition as control function "ENERGY-RATE". The procedure is to first differentiate the cumulative mass source, given by control function "INT-MASS-SOURCE", to define the source rate in control function "POOL-RATE", and then multiply this rate by the specific enthalpy of the pool (material 1) in control volume BOUNDARY. See the CF Package Users' Guide for more information on specific CF input.

The final example involves fluid entering the volume CV300 at an elevation of 10.0 m, where a superheated pool is allowed to flash to produce steam and fog in that volume whenever the pool elevation is below the entrance elevation. (It is assumed that an appropriate READ external data file "EDF1" has been provided to define the rates.)

```
CV_ID  CV300
CV_SOU 2 !N, SourceInfo    1 WM     RATE  EDF EDF1  1
          2 WE     RATE  EDF EDF1  4  10.0
          3 MASS   RATE  EDF EDF1  2  FOG
          4 MASS   RATE  EDF EDF1  3  H2O-VAP
          5 AE     RATE  EDF EDF1  5
```

7 CVH Package Output

Each printed edit generated by the CVH package in MELGEN and MELCOR contains a snapshot description of the state of all control volumes, with the output organized in tabular form. Many column headings are abbreviated, but most are relatively clear—particularly when it is understood that LIQ refers to the pool, and VAP refers to the atmosphere. One frequent source of confusion is that, in the CVH output, "VOID FRACTION" and "QUALITY" refer to the partition of volume and of water mass, respectively, between pool and atmosphere. The vapor bubble content of the pool may be inferred from the difference between the swollen liquid level "S-Z-POOL" and the collapsed liquid level "C-Z-POOL".

The edit includes summaries of the integrated sources in each control volume and of the current partition of volume between hydrodynamic ("MATL") and virtual volume.

A description of the nodalization, including a list of all flow paths connected to each volume, will be found in the "FLOW PATH TIME INDEPENDENT DATA EDIT" generated by the FL package in MELGEN and in the first edit for each MELCOR run.

Users are strongly advised to check the initial edit generated by MELGEN—in addition to the contents of the diagnostic file (MEGDIA)—before proceeding with a calculation.

8 Diagnostics and Error Messages

Diagnostic messages may be written either by MELGEN or by MELCOR. Messages from MELGEN ordinarily report errors or inconsistencies in input; problems encountered while completing the definition of initial conditions may also be reported. If an error is encountered, no restart file will be written; errors must be corrected before MELCOR can be run.

Messages from MELCOR report problems encountered during the time advancement of a calculation. If an error is severe, the calculation will be stopped, and a restart dump will be written corresponding to conditions at the beginning of the fatal timestep.

A user input option, CVHTRACE, turns on a detailed, step-by-step tracing of the numerical solution of the hydrodynamic equations, with information written to the diagnostic file (MELDIA). If the calculational performance of the CVH package is poor, this information may be useful in determining the cause.

If any package requires a timestep below the user-defined minimum value, the MELCOR executive will stop the calculation. (The SOFTDTMIN record in the EXEC Package Users' Guide allows this treatment to be modified.) When this occurs, the executive will repeat the last attempt with CVHTRACE turned on. In most such cases, the unacceptable timestep cut will have been demanded by the CVH package. In a large fraction of these, this will be found to be a result of unacceptable boundary conditions imposed by other packages. The output from CVHTRACE is often helpful in identifying the underlying cause.

8.1 Input and Execution Messages

Messages resulting from errors in record formats or unrecognized records in the user input to MELGEN include the record identifier and sufficient information to allow the user to identify the error. Typical errors include failure to supply all required input records and incomplete specification or over-specification of the initial thermodynamic state of a control volume. Inconsistencies with MEGLEN input to other packages, including failure to correctly define any tabular or control functions or external data files required to define sources, will also be identified.

Input processing continues after an error has been found in order to identify multiple errors, but MEGLEN will not write an initial restart until all input errors have been eliminated. The

default assumptions used to allow processing to continue may themselves lead to additional error messages. If one of a group of input diagnostics appears irrelevant to the actual input, the proper strategy is to make as many corrections as possible and rerun MELGEN; the other error messages will be eliminated or clarified.

Error or warning messages may be issued during execution of MELCOR for any number of reasons. These include attempts by another package or a CVH sink to remove more mass or energy than is contained in the associated control volume, improper relocation of virtual volume by other packages, and errors in tabular or control functions or external data files defining CVH sources or the properties of time-specified volumes. Execution will continue with a reduced system timestep, when it makes sense to do so; the request for a system fallback will be noted.

In some cases, the thermodynamics routines will transfer energy between the pool and the atmosphere of a control volume, rather than return a subfreezing temperature for one of them. Such occurrences are noted; a problem with input is usually involved. A special case is the discharge of a high-pressure accumulator where the gas temperature will fall below freezing under adiabatic expansion. Although MELCOR thermodynamics does not permit temperatures below 273.15 K, this energy transfer allows the calculation to continue with an isothermal expansion of gas after the freezing point is reached. (After fifty occurrences of such energy transfer, further warnings are suppressed.)

Coding of the CVH package includes a number of messages warning of apparent logic errors, with the text identifying them as such. The appearance of a message of this type probably involves a coding error in MELCOR; the code development team should be informed.

8.2 CVHTRACE Option

Inclusion of the CVHTRACE record in MELCOR input enables tracing of the calculations through which the CVH package advances. The result will be a large number of informative messages written to the diagnostic file (MELDIA). These messages follow very closely the description of the CVH solution algorithm (see the CVH/FL Reference Manual).

The volume of messages, while not nearly so large as was produced by the obsolete MELCOR DEBUG option, is large enough that it should not be used for more than a few cycles. Therefore, its activation is not recorded in the restart file. Tracing is active if, and only if, the CVHTRACE record is present in MELCOR input for the current run.

The CVHTRACE option evolved as a partial substitute for problem debugging using system-specific debugging software such as dbx on UNIX systems. In essence, it automates the tracing and examination of variables habitually used by the code developers to investigate problems in calculations. The output assumes that the user knows a little about MELCOR numerics, but the output is not intended to be excessively cryptic.

Use of the CVHTRACE option should aid in identification of problem areas in a calculation. If a calculation is proceeding smoothly but with very small timesteps, the information provided by the message

NEXT TIME STEP LIMITED TO n.nnnnEeee S BY <something>

should identify the reason. If the timesteps are limited by the material Courant condition in a volume, it may be appropriate to combine this volume with a larger neighbor to ease the constraint. If the limit is the rate of change in temperature or pressure in a volume, the cause may be simply a rapid transient in conditions after which the timestep should recover. If the system is under-damped, such a transient may persist long after the initial perturbation has passed. This may reflect input data that places too little frictional loss in the flow paths. If no such transient should be occurring, the problem may be a rapidly fluctuating boundary condition imposed on that volume by some other package. It may be possible to modify input to the other package to reduce the fluctuation.

If, on the other hand, each advancement is found to involve many iterations with repeated subcycle and timestep fallbacks, it is likely that one or two volumes or flow paths will appear repeatedly in the lists of "last-to-converge" or "failed-to-converge". Attention should then be focused on these volumes and/or flow paths and their relationships to the core, heat structures, and other elements of MELCOR modeling. Simple changes in nodalization—particularly in junction opening heights, inertial lengths, momentum exchange lengths, and loss coefficients in the flow path input—may be sufficient to eliminate the problem. Another strategy is to restrict the timestep using TIMEn records in the EXEC package input to a value commensurate with the successful subcycle advancements.

If convergence problems are seen for a flow path that is ultimately found to be choked, it may be appropriate to increase the loss coefficients for that path and/or to increase the momentum exchange length to reduce the change in velocity calculated before choking is considered.

If a volume is repeatedly mentioned in association with the message,

***ADVANCEMENT FAILED: THERMO ERROR

any sources or sinks in that volume, from CVH input or other packages, should be closely examined. The most common cause of such an error is a source in the volume that drives the volume out of the range of validity of the equation of state.

Throughout the process, however, the user should remain aware that CVHTRACE can only display symptoms. The underlying causes of a problem may lie in areas of the calculation that are apparently experiencing no difficulties at all.

An outline of the contents of the diagnostic file is given below for a case where the CVHTRACE option is enabled. Four character strings in the messages should be noted as potential search patterns for the diagnostic file:

"===" Marks the start of a MELCOR timestep,

"…" Marks the start of an attempted step (full or subcycle) in CVH,

"---" Marks the successful completion of such an advancement, and

"***" Marks a problem, a point where a calculation had to be repeated, or a point where a compromise was required (for example, the user input minimum timestep overrode the value desired by CVH).

Indentation is used as a further aid to scanning the contents of the diagnostic file. This indentation is duplicated in the example messages shown below.

8.2.1 Main Advancement

The start of each MELCOR timestep is marked by the message

===CVH ADVANCEMENT FOR CYCLE nnnnnn DT = x.xxxxEeee S

If sinks (negative sources) of mass and/or energy specified by CVH input require more mass or energy than is present in one or more volumes, the problem volumes are listed, followed by the message

***FALLBACK: EXCESSIVE MASS OR ENERGY SINK

and a MELCOR fallback is performed. Tracing will begin again at "===CVH ADVANCEMENT …" with a smaller value of DT.

The start of each attempted CVH (sub)cycle is marked by the message

…ATTEMPT (SUB)STEP OF x.xxxxEeee S, PRESSURE ITERATION n

Pressure iteration refers to the (re)linearization of terms involving new volume pressures in the acceleration of flows and (re)solution of the resulting flow equations.

Diagnostic messages are generated during solution of the flow equations. See Section 8.2.2 for a detailed discussion. If the solution fails, a fallback within CVH is executed, and tracing will generally begin again at "…ATTEMPT (SUB)STEP …" with a smaller substep. However, if the substep has become too small, the message

***FALLBACK: SUBCYCLE STEP TOO SMALL COMPARED TO DT

is generated, and a MELCOR fallback is executed. Tracing will begin again at "===CVH ADVANCEMENT …" with a smaller value of DT.

Once the flow solution is complete, mass and energy moves are computed, and the (material) Courant limit is calculated and noted as

COURANT LIMIT OF x.xxxxEee S SET BY VOLUME NameCV

(A further message is provided if this limit is exceeded, but should not appear given current coding.)

The new volume pressures are computed from the new masses and energies. If the discrepancy from the linearly-projected pressure is excessive, the offending volumes are noted, and the message

***ADVANCEMENT FAILED: PRESSURE CONVERGENCE FAILURE

is generated. Tracing begins again at "…ATTEMPT (SUB)STEP …" either as a new pressure iteration or if the maximum number of iterations has been reached, with a decreased substep size following an internal fallback.

If errors occur in evaluation of the equation of state, the problem volumes are noted, the message

***ADVANCEMENT FAILED: THERMO ERROR

is written, an internal fallback is executed, and tracing begins again at "…ATTEMPT (SUB)STEP …" with a decreased substep size.

Messages are coded for a possible "water-pack–fix-up", but this is not enabled in current versions of MELCOR.

When a substep is completed successfully, the net advancement to that point is noted as

---ADVANCEMENT SUCCESSFUL THROUGH x.xxxxEeeee S

The possibility of choking is then evaluated. If it is detected, the step is repeated with the message

***REPEAT FOR CHOKING IN FLOW PATHS NameFL NameFL …

When the advancement reaches the end of the MELCOR timestep, the net changes during the step are examined, and if all are acceptable, a proposed new timestep is estimated. This is noted as

NEXT TIME STEP LIMITED TO n.nnnnEee S BY <something>

where <something> may be a change in pressure or temperature (using limits defined by sensitivity coefficients in array 4400) in some volume (the volume name will be noted), the Courant limit (the relevant volume name has already been noted), or the estimated accuracy of some solution procedure. This will be followed by the message

***MINIMUM TIMESTEP OVERRIDES LIMIT

if appropriate.

If some change is excessive (again based on sensitivity coefficients in array 4400), however, a message is issued:

***EXCESSIVE CHANGE IN <something>, VOLUME NameCV

where <something> may be POOL TEMPERATURE, ATMOSPHERE TEMPERATURE, or PRESSURE. It will generally be followed by the message,

***FALLBACK: CHANGE DURING TIMESTEP TOO GREAT

and the entire MELCOR step will be repeated with a decreased value of DT. However, if the step is already within a factor of 2 of the minimum, the calculation will continue with the warning message

***MINIMUM TIMESTEP PREVENTS FALLBACK

8.2.2 Flow Solution

Solution of the flow equation is iterative, because donor properties and friction terms are nonlinear functions of the flow velocities. When the iteration converges, the message

VELOCITIES CONVERGED IN CVHMOM ON ITERATION nn

is written. If iteration was required, this is followed by either or both of the messages

LAST VELOCITIES TO REVERSE INCLUDE NameFL/c NameFL/c <etc>

LAST VELOCITIES TO CONVERGE INCLUDE NameFL/c NameFL/c <etc>

where NameFL is a flow path name and c is P for the pool or A for the atmosphere. If the number of flows in either list exceeds 10, the list is truncated. "Reversal" is with respect to the previous iteration (so that donoring must be switched). Convergence requires a change in magnitude from the previous iteration within the limits set by sensitivity coefficients in array 4401.

In some cases, the volumetric flows converge even though the velocities do not, because of iteration-to-iteration changes in the flow-path void fraction. If this is the case, the messages will be

VOLUME FLOWS CONVERGED IN CVHMOM ON ITERATION nn

UNCONVERGED VELOCITIES INCLUDE NameFL/c NameFL/c <etc>

If, after a number of iterations defined by a sensitivity coefficient in array 4401, neither velocities nor flows have converged, but the unconverged ones do not contribute significantly to the projected new pressure, the iteration is terminated with the message

PRESSURES CONVERGED IN CVHMOM ON ITERATION nn

and either or both of the messages

REVERSED VELOCITIES INCLUDE NameFL/c NameFL/c <etc>

UNCONVERGED VELOCITIES INCLUDE NameFL/c NameFL/c <etc>

(The default is not to allow acceptance of a velocity solution on this basis.)

The solution may fail with either the message

***VELOCITY ITERATION FAILED IN CVHMOM

or the message

***FLOW MATRIX IN CVHMOM 'SINGULAR' ON ITERATION nn

The maximum number of iterations is set by a sensitivity coefficient in array 4401.

Appendix 1

MELCOR 2.0	PTFRead	Comments
CVH-ATM-FR(NameCV) NameCV – the control volume name.	CVH-ATM-FR.n n – the control volume index.	Atmosphere (non-pool) volume fraction in control volume NameCV (n).
CVH-CLIQLEV(NameCV) NameCV – the control volume name.	CVH-CLIQLEV.n n – the control volume index.	Collapsed liquid elevation in control volume NameCV (n).
CVH-CPUT	CVH-CPUT	Total CPU usage by the RUN (time-advancement) portion of the CVH package.
CVH-CPUE	CVH-CPUE	CPU usage for edit in the RUN portion of the CVH package.
CVH-CPUC	CVH-CPUC	CPU usage for calculations in the RUN portion of the CVH package.
CVH-CPUR	CVH-CPUR	CPU usage to process the restart file in the RUN portion of the CVH package.
CVH-E(NameCV,NameMat) NameCV – the control volume name; NameMat – the material name. The choices allowed are: ALL – for all materials; NCG material name.	CVH-E.n CVH-E.m.n n – the control volume index; m – the material number.	Specific internal energy of material NameMat (m) in control volume NameCV (n) or specific internal energy in control volume NameCV if NameMat = ALL
CVH-ECV (NameCV,NameMat) NameCV – the control volume name; NameMat – the material name. The choices allowed	CVH-ECV.n CVH-ECV.m.n n – the control volume index; m – the material number.	Total internal energy of material NameMat (m) in control volume NameCV (n) or total internal energy in the control volume if NameMat = ALL.

MELCOR 2.0	PTFRead	Comments
are: ALL – for all materials; NCG material name.		
CVH-H(NameCV,NameMat) NameCV – the control volume name; NameMat – the material name. The choices allowed are: ALL – for all materials; NCG material name.	CVH-H.n CVH-H.m.n n – the control volume index; m – the material number.	Specific enthalpy of material NameMat (m) in control volume NameCV (n) or specific enthalpy in the control volume if NameMat = ALL.
CVH-LIQLEV(NameCV) NameCV – the control volume name.	CVH-LIQLEV.n n – the control volume index.	Swollen liquid elevation in control volume NameCV (n).
CVH-MASS (NameCV,NameMat) NameCV – the control volume name; NameMat – the material name. The choices allowed are: ALL – for all materials; NCG material name.	CVH-MASS.n CVH-MASS.m.n n – the control volume index; m – the material number.	Mass of material NameMat (m) in control volume NameCV (n) or mass of material in the control volume if NameMat = ALL.
CVH-P(NameCV) NameCV – the control volume name.	CVH-P.n n – the control volume index.	Pressure of control volume NameCV (n).
CVH-PPART (NameCV,NameMat) NameCV – the control volume name. NameMat – the material name. The choices allowed are: ALL – for all materials; NCG material name.	CVH-PPART.m.n n – the control volume index; m – the material number.	Partial pressure of material NameMat (m) in control volume NameCV (n).
CVH-PSAT(NameCV,key) NameCV – the control volume name. The choices allowed for key	CVH-PSAT(TLIQ).n CVH-PSAT(TVAP).n n – the control volume index.	Saturation pressure of water at the atmosphere (key = ATM) or at the pool (key = POOL) temperature in control volume NameCV

MELCOR 2.0	PTFRead	Comments
are: key = ATM for atmosphere; key = POOL for pool.		(n).
CVH-QUALITY(NameCV) NameCV – the control volume name.	CVH-QUALITY.n n – the control volume index.	Quality (non-pool, non-fog mass fraction) in control volume NameCV (n).

CVH-RHO(NameCV,key)
 	
NameCV – the control volume name.
The choices allowed for key are:
key = ALL for all materials
key = ATM for atmosphere;
key = POOL for pool;
key = NameMat for material name.
	CVH-RHO.n
	CVH-RHO.m.n
	n – the control volume index;
	m – the material number.
		Mass density of materials in control volume NameCV (n).

CVH-TLIQ(NameCV) NameCV – the control volume name.	CVH-TLIQ.n n – the control volume index.	Pool temperature in control volume NameCV (n).
CVH-TOT-E(NameMat) NameMat – the material name. The choices allowed are: NCG material name.	CVH-TOT-E.m m – the material number.	Total internal energy of material NameMAT (m) in the CVH package.
CVH-TOT-M(NameMat) NameMat – the material name. The choices allowed are: NCG material.	CVH-TOT-M.m m – the material number.	Total mass of material NameMat (m) in the CVH package.
CVH-TSAT(NameCV,key) NameCV – the control volume name. The choices allowed for key are: key = A for atmosphere; key = P for pool.	CVH-TSAT(P).n CVH-TSAT(A).n n – the control volume index.	Water saturation temperature at the pressure (key = P) or water saturation temperature at the partial pressure of water vapor in the atmosphere (key = A) of control volume NameCV (n).
CVH-TVAP(NameCV)	CVH-TVAP.n	Atmosphere temperature in control volume NameCV

MELCOR 2.0	PTFRead	Comments
NameCV – the control volume name.	n – the control volume index.	(n).
CVH-VEL(NameCV,key) NameCV – the control volume name. The choices allowed for key are: key = ATM for atmosphere; key = POOL for pool.	CVH-VELLIQCV.n CVH-VELVAPCV.n n – the control volume index.	Velocity of the atmosphere (key = ATM) or the pool (key = POOL) in control volume NameCV (n).
CVH-VIRVOL(NameCV,key) NameCV – the control volume name. The choices allowed for key are: key = ATM for atmosphere; key = POOL for pool.	CVH-VIRVOLIQ.n CVH-VIRVOVAP.n n – the control volume index.	Virtual volume in the atmosphere (key = ATM) or the pool (key = POOL) in control volume NameCV (n).
CVH-VOID(NameCV,key) NameCV – the control volume name. The choices allowed for key are: key = ATM for atmosphere; key = POOL for pool.	CVH-VOID-P.n CVH-VOID-T.n n – the control volume index.	Void (bubble) volume fraction in the pool (key = POOL) of control volume NameCV (n) or total void (non-liquid) volume fraction, including the atmosphere (key = ATM) and bubbles in the pool.
CVH-VOLFOG(NameCV) NameCV – the control volume name.	CVH-VOLFOG.n n – the control volume index.	Volume of fog in control volume NameCV (n).
CVH-VOLLIQ(NameCV) NameCV – the control volume name.	CVH-VOLLIQ.n n – the control volume index.	Swollen volume of the pool (including any vapor bubbles) in control volume NameCV (n).
CVH-VOLVAP(NameCV) NameCV – the control volume name.	CVH-VOLVAP.n n – the control volume index.	Volume of the gaseous atmosphere in control volume NameCV (n).
CVH-X(NameCV,NameMat) NameCV – the control volume name. NameMat – the material name. The choices allowed	CVH-X.m.n n – the control volume index; m – the material number.	Mole fraction of gaseous material NameMat (m) (m $\geq$ 3) in the atmosphere of control volume NameCV (n).

MELCOR 2.0	PTFRead	Comments
are: ALL – for all materials; NCG material.		

Decay Heat (DCH) Package
Users' Guide

The MELCOR Decay Heat (DCH) package models the decay heat power resulting from the radioactive decay of fission products. Decay heat is evaluated for reactor core material and for suspended or deposited aerosols and gases. MELCOR couples thermal-hydraulic processes and fission product behavior during the calculation.

Both the radionuclides present in the reactor at the time of the accident and the radionuclide daughter products contribute to the decay heat. In the calculation of decay heat, MELCOR does not explicitly treat each decay chain, since detailed tracking of radionuclide decay chains would be too costly. When the radionuclide package is active, the decay heat is calculated for each radionuclide class by using pre-calculated tables from ORIGEN program runs. If the radionuclide package is not active, the whole-core decay heat is computed from one of several possible user-defined calculations.

This users' guide describes the input to the DCH package, including a brief description of the models employed, the input format, sample input, discussion of the output, sensitivity coefficients, plot variables, and control variables. Details on the models can be found in the DCH Package Reference Manual.

Contents

CONTENTS **2**

1 INTRODUCTION **4**

1.1 Explicit definition of operating power 4

1.2 Activity Calculations (BONUS code) 4

2 USER INPUT **5**

2.1 MELGEN Input 5

2.1.1 General Input 5

DCH_RCT – Reactor type 5
DCH_SHT – Reactor Shutdown Time 7

2.1.2 Whole-Core Input 8

DCH_OPW – Reactor Operating Power 8
DCH_DPW – Decay Power of Nuclides 8
DCH_FPW – Fission and Decay Power of Nuclides 9
DCH_OPT – Reactor Operating Time 9
DCH_NFA – Number of Fissions per Initial Fissile Atom 10

2.1.3 Radionuclide Class Input 10

DCH_EL – Element Name and Time, Decay Heat Data 11
DCH_CL – Radionuclide Class Name and Elements in Class 12
DCH_NRM – Class Decay Heat Normalization Flag 13

2.1.4 Activity Calculations (BONUS) 14

DCH_ACT – Activity calculation options 14
DCH_SUR Output activity data 15

2.2 DCH Example Input for Activity Calculation 16

2.3 MELCOR Input 16

2.4 Additional Library Input Files (BONUS) 16

2.4.1 File 'FissProd.in' (properties of fission products) 16

2.4.2 File 'FPChains.in' (structure of isobaric chains) 17

2.4.3 File ' InitInventory.in' (structure of initial inventory file) 19

3 SENSITIVITY COEFFICIENTS **20**

3.1 DCH Sensitivity Coefficients 20

3200 – Multiplier for ANS Decay Heat Curve 20
3201 – Energy per Fission for Nuclides 20
3202 – Times for ANS Decay Heat Power Tables 21

3203 – Decay Heat Powers for ANS Decay Heat Curve 21
3204 – Neutron Capture Correction factors for ANS Decay Heat Curve 21
3205 – Parameters for Actinide Decay Heat Calculation 21
3210 – Multiplier for all ORIGEN Elemental Decay Heat Curves..... 22
3211 – Multipliers for Individual ORIGEN Elemental Decay Heat Curves 22
3212 – Fraction of Fuel Equilibrium Cycle Elapsed 23
DCH_SC – DCH Sensitivity Coefficients Input Record..... 23

4 PLOT VARIABLES.....24

5 CONTROL FUNCTION ARGUMENTS 24

6 EXAMPLE INPUT.....25

6.1 General Input 25
6.2 Whole-Core Input 25
6.3 Radionuclide Class Input 25
6.4 Sensitivity Coefficients Input 26

7 EXAMPLE OUTPUT.....27

7.1 MELGEN Output 27
7.1.1 Standard DCH Output 27
7.2 Standard MELCOR Output..... 30
7.3 MELCOR Activity Output..... 30

8 DIAGNOSTICS AND ERROR MESSAGES.....32

1 Introduction

The MELCOR Decay Heat (DCH) package models the heating from the radioactive decay of fission products. Decay heat power is evaluated for the fission products assumed to reside in reactor core materials, cavity materials, and in suspended or deposited aerosols and vapors. Decay heat power levels as a function of time are supplied as a utility function within MELCOR that may be called by other phenomenological packages. The DCH package is not involved in the calculation of fission product transport or chemical interactions. These processes are calculated by the RadioNuclide (RN) package (see the RN Package Reference Manual).

Both the radionuclides present in the reactor core and/or cavity from the time of reactor shutdown and the radionuclide daughters from decay contribute to the total decay heat power. In the calculation of decay heat power, the DCH package does not explicitly treat decay chains. Detailed tracking of radionuclide decay chains was seen as computationally costly and too detailed for MELCOR. Instead, when the RN package is active, elemental decay heat power information based on ORIGEN calculations is summed into the RN class structure, as described in the DCH Package Reference Manual.

There are also several options for calculating decay heat power when the RN package is not active (that is, when tracking of fission products is not desired). These are called whole-core calculations in the DCH package, although they may be applied to cavity inventories of melt debris as well.

1.1 Explicit definition of operating power

In many applications, the decay heat levels and/or the initial RN inventories in fuel are calculated based on the operating power of the reactor. In MELCOR 1.8.5 and earlier versions, the operating power was inferred from input on the DCH_FPW record (or equivalent defaults, even if the ANS option for whole-core power was not used). This led to a number of misunderstandings and resulted in calculations where the decay power and/or the fission product inventory was not as expected. MELCOR 1.8.6 and later versions require explicit input of the operating power on a new input record.

1.2 Activity Calculations (BONUS code)

The MELCOR code has a limited capability to perform activity calculations utilizing the RN and DCH packages. Consistent with the level of accuracy inherent in an integral severe accident code, BONUS, a simplified code for tracking radioactive decay has been implemented in to MELCOR by IBRAE for this purpose. The version of BONUS implemented into MELCOR only evaluates the concentration of fission products after reactor shutdown. The user supplies input for describing decay chains and radionuclide properties (half-life times, decay energies, direct and cumulative yields). The one limitation on decay chains is that they be unidirectional, preventing their cycling (though for actinides this restriction is not imposed). Also, this implementation does not account for transformation between chemical classes. The chemical class inventory is provided at shut-down and no movement between classes due to decay are considered.

2 User Input

2.1 MELGEN Input

2.1.1 General Input

These input records describe data used by both the whole-core and class decay heat calculations.

DCH_RCT – Reactor type

Optional

This record enables the user to specify the type of reactor. This will affect the default power levels and, for the radionuclide class calculation, the radionuclide mass inventories. The valid options and the default are core-model specific; values given here are for the light water reactor (LWR) Core (COR) package. Users of an alternate core package should consult the documentation for that package. Note that the default type of pressurized-water reactor (PWR) is inconsistent with the default reactor type in the COR package on record COR_RT; one or the other should be changed by the user. If this record is present, the value–default or not–must be input.

(1) REACTP

Character string selecting reactor type:

(a) PWR

Use PWR mass inventories for radionuclide class calculation and set default power to 3412 MW.

(b) BWR

Use boiling-water reactor (BWR) mass inventories for radionuclide class calculation and set default power to 3578 MW.

(type = character * 3, default = PWR)

Example

```
DCH_RCT   BWR
```

If the user has specified that this calculation will include an activity calculation using the BONUS code (IACTIV=ON on DCH_ACT record) then the user may also specify the filename and data block with initial inventories. In this case, the input for this record is modified with additional required information:

For IACTIV = 1 or ON and INPFORM = 0 or OLD on DCH_ACT record, then

(1) REACTP

Character string selecting reactor type:

(a) PWR, use PWR mass inventories for radionuclide class calculation and set default power to 3412 MW.
(b) BWR, use BWR mass inventories for radionuclide class calculation and set default power to 3578 MW.
(type = character * 3, default = 'PWR')

(2) FILE

Character string selecting the name of the file containing initial inventories.

(type = character*80, default = 'InitInventory.in')

(3) BLOCK

Character string selecting the name of the data block containing initial inventory values in file "FILE".

(type = character*80, default = 'REACTOR_PWR')

For IACTIV = 1 or ON and INPFORM = 1 or NEW on DCH_ACT record, then

(1) FILE

Character string selecting the name of the file containing initial inventories.

(type = character*80, default = 'InitInventory.in')

(2) BLOCK

Character string selecting the name of the data block containing initial inventory values in file "FILE".

(type = character*80, default = 'REACTOR_PWR')

Example

Old MELCOR input:

```
DCH_RCT   PWR "initInventory.in"  "REACTOR_PWR"
```

New MELCOR input:

```
DCH_RCT "initInventory.in"   "REACTOR_BWR"
```

If INPFORM = OLD, the type of reactor can be specified, which will affect the default power levels and, for the radionuclide class calculation, the radionuclide mass inventories. The valid options and the default are core-model specific; values given here are for the LWR COR package. Users of an alternate core package should consult the documentation for that package. Note that if the default type of 'PWR' is inconsistent with the default reactor type in the COR package on record COR_RT then one or the other should be changed by the user.

If a filename is specified (required for activity calculations, IACTIV = 1 or ON on DCH_ACT record), then this file must be present in the same directory with input file. The default initial inventory file (see section 2.4.3) contains several blocks for different reactor types and burn-ups. The user must specify the name of block to be used (default value is REACTOR_PWR). If the user has specified a new data block in the initial inventory file, the name of this block should start with "REACTOR_" record. For example, "REACTOR_USERINPUT".

DCH_SHT – Reactor Shutdown Time

Optional

This record defines the problem time at which reactor shutdown occurs for purposes of decay heat computation.

(1) SHFLG

Flag for reactor shutdown time:

(a) CF

Use ISHTCF as trip control function for decay heat.

(b) TIME

Use TMSHUT as input for reactor shutdown time.

(type = integer, default = TIME)

If SHFLG = CF, then

(2) ISHTCF

Reactor shutdown (logical valued) control function name used to determine the reactor shutdown time. During computation the reactor shutdown time is set to the current problem time whenever the control function value changes from .FALSE. to .TRUE.

(type = integer, default = none, units = none)

Otherwise (SHFLG = TIME), TMSHUT field is required.

(2) TMSHUT

Reactor shutdown time.

Required if SHFLG = TIME.

(type = real, default = 0.0, units = s)

Example

```
DCH_SHT TIME  0.0
DCH_SHT CF   'Shutdown CF'
```

2.1.2 Whole-Core Input

These input records describe data used by the whole-core decay heat calculations.

DCH_OPW – Reactor Operating Power

Required

This record defines the operating power of the reactor before shutdown. The value is used in calculation of fission product inventories and decay power.

(1) OPRPOW

Reactor operating power (thermal).

(type = real, default = none, units = W)

Example

```
DCH_OPW  3400.0E6
```

DCH_DPW – Decay Power of Nuclides

Optional

This record enables the user to specify the calculation to be used for the whole-core decay heat calculation.

(1) WCTYPE

Type of whole-core decay heat (W) calculation, character string of four possible forms:

(a) ORIGEN

Use summation of default ORIGEN radionuclide decay heats (note that the value will not be affected by changes to elemental decay power tables input on DCH_EL input records)

(b) ANS

Use ANS standard calculation

(c) CF

Use control function for decay heat (W)

(d) TF

Use tabular function as table of whole-core decay heat (W) as function of time since shutdown in seconds

(type = character * 6, default = ORIGEN)

If WCTYPE = CF or TF,

(2) TFCFNAME

Name of control function used if WCTYPE = CF or

Name of tabular function used if WCTYPE = TF

(type = character * 16, default = none)

Example

```
DCH_DPW TF   TF50
DCH_DPW ANS
```

DCH_FPW – Fission and Decay Power of Nuclides

Optional

This record defines the total thermal operating power due to the nuclides U-235, Pu-239, and U-238 (from both fission and decay heat). These data are used only if WCTYPE on card DCH_DPW is ORIGEN or ANS. The defaults are core-model specific; values given here are for the LWR COR package. These defaults are taken from a SANDIA-ORIGEN run giving the powers averaged over an equilibrium fuel cycle. The defaults for total power are 3412 MW (PWR) and 3578 MW (BWR). Users of an alternate core package should consult the documentation for that package.

(1) U235P

Power due to thermal fission of U-235.

(type = real, default = 2.2086E9 (PWR), 2.316E9 (BWR), units = W)

(2) PU239P

Power due to thermal fission of Pu-239.

(type = real, default = 1.0598E9 (PWR), 1.1114E9 (BWR), units = W)

(3) U238P

Power due to fast fission of U-238.

(type = real, default = 1.436E8 (PWR), 1.506E8 (BWR), units = W)

DCH_FPW (DCHFPOW): there is no positivity checking in the code for variables U235P, PU239P, and U238P.

Example

```
DCH_FPW 0.2316E+10 0.1197E+10 0.1506E+09 ! Decay power U-235, Pu-239, U-238
```

DCH_OPT – Reactor Operating Time

Optional

This record defines the irradiation time, the length of time the reactor fuel has been undergoing fission. A constant fission rate is assumed during this time. The default value is 0.8 x 2 years, or 584 days, chosen to match the ORIGEN run with an average in-core time of 2 years for the fuel at the end of the equilibrium cycle with an 80% capacity factor. These data are used only if WCTYPE on card DCH_DPW is ANS.

(1) OPRTIM

Reactor operating time.

(type = real, default = 5.05E7, units = s)

Example

```
DCH_OPT 5.05E7
```

DCH_NFA – Number of Fissions per Initial Fissile Atom

Optional

This factor is used as in the calculation of the neutron capture correction factor *G(t)* (see Section 3.2 of the DCH Package Reference Manual). The default is computed from SANDIA-ORIGEN data for a representative fissile material inventory. These data are used only if WCTYPE on card DCH_DPW is ANS.

(1) PSINC

Number of fissions per initial fissile atom.

(type = real, default = 0.713, units = fissions/atom)

Example

```
DCH_NFA 0.637
```

2.1.3 Radionuclide Class Input

These input records describe data used for the radionuclide class decay heat calculations. The grouping of elements into RN classes is described in detail in Section 2.2 of the DCH Package Reference Manual. The valid names for default elements input on records DCH_EL and DCH_CL are listed below. Only those listed without parentheses in Table 2.1 of the DCH Package Reference Manual have decay heat tables in MELCOR. Only those elements (and their isotopes) marked in bold, **red** type will activities be calculated.

Table 2.1 Valid Names for Default Elements

'H',	'HE',	'LI',	'BE',	'B',	'C',	'N',	'O',	'F',	'NE',
'NA',	'MG',	'AL',	'SI',	'P',	'S',	'CL',	'AR',	'K',	'CA',

'SC',	'TI',	'V',	'CR',	'MN',	'FE',	'CO',	'NI',	'CU',	'ZN',
'GA',	'GE',	**'AS',**	**'SE',**	**'BR',**	**'KR',**	**'RB',**	**'SR',**	**'Y',**	**'ZR',**
'NB',	**'MO',**	**'TC',**	**'RU',**	**'RH',**	**'PD',**	**'AG',**	**'CD',**	**'IN',**	**'SN',**
'SB',	**'TE',**	**'I',**	**'XE',**	**'CS',**	**'BA',**	**'LA',**	**'CE',**	**'PR',**	**'ND',**
'PM',	**'SM',**	**'EU',**	**'GD',**	'TB',	'DY',	'HO',	'ER',	'TM',	'YB',
'LU',	'HF',	'TA',	'W',	'RE',	'OS',	'IR',	'PT',	'AU',	'HG',
'TL',	'PB',	'BI',	'PO',	'AT',	'RN',	'FR',	'RA',	'AC',	'TH',
'PA',	**'U',**	**'NP',**	**'PU',**	'AM',	'CM',	'BK',	'CF',	'ES',	'FM',
'WT'(water)				'CC'(concrete)					

It should be noted that BONUS deals with FPs and some actinides but not with structural materials. Moreover, only FPs with a cumulative yield more than 0.1% with a half-life more than about a minute are considered.

DCH_EL – Element Name and Time, Decay Heat Data

Optional.

If it is absent, all elements in class are default.

This record allows the user to define a new element or to redefine a default element for purposes of decay heat calculation. If the user inputs a default element, the user-defined element decay heat data are used instead of the default data.

(1) ELMNAM

User-input element name. May be a default element or a new name.

(type = character*2)

(2) ELMMAS

Mass inventory of element ELMNAM in reactor at shutdown.

(type = real, default = none, units = kg)

The last optional fields define the pairs of decay heat data for element ELMNAM. The first field in a pair is the time after shutdown, and the second field is the decay heat power per unit mass at that time. The decay heat times must be in ascending order.

(3) NMPAIR

Number of data pairs.

(type = integer, default = none, units = none)

The following data are input as a table with length NMPAIR:

(1) NPAIR

Data pair index.

(type = integer, default = none, units = none)

(2) TIME

Time after shutdown.

(type = real, default = none, units = s)

(3) DCHEAT

Decay heat power per unit mass for this element at time TIME.

(type = real, default = none, units = W/kg)

Example

```
DCH_EL  AS    0.23E-03   2
      !N      TIME             DHEAT
       1      0.0               0.0
       2      0.95E+06          0.0
```

The user must define all classes which will be used in the radionuclide calculation on DCH_CL records.

DCH_CL – Radionuclide Class Name and Elements in Class

Required

This record is required for all classes (default or not) used in the problem. If one of the default classes (listed in Table 2.1 of the DCH Package Reference Manual) is used, the user must input the character string DEFAULT on field FLAGCLS. If the user defines a new class or redefines a default class for purposes of decay heat calculation, then FLAGCLS = USER and the additional data fields must be input.

If user wants to calculate activity, the names of elements must correspond to chemical element names listed in the Table 2.1 (in bold type), otherwise the activity will not be calculated!

(1) RDCNAM

User-input class name.

(type = character*16)

(2) FLAGCLS

Flag for class type.

(a) DEFAULT

Default class. No additional data are input.

(b) USER
New or redefined class.

(type = character * 8)

If FLAGCLS = USER,

(3) NCLELM
Number of elements.

(type = integer, default = none)

The following data are input as a list with length NCLELM:

(1) NELM
Element index.

(type = integer)

(2) CLSELM
Name of element in class. The element name may be a user-defined element or a default element.

(type = character*2)

Example

```
DCH_CL 'XE'  DEFAULT
DCH_CL 'CS'  DEFAULT
DCH_CL 'BA'  DEFAULT
DCH_CL 'CSI' USER   1
   1 'CI'   ! New Element 'CI' defined in DCH_EL table
```

DCH_NRM – Class Decay Heat Normalization Flag

Optional

This record allows the user to have the radionuclide class decay heat power normalized to the whole-core decay heat power. If the normalization flag is set to YES, the calculated decay heat for each class (whether default or user-defined) will be multiplied by the ratio of the sum of all the class decay heats and the whole-core decay heat. Thus, the sum of all the class decay powers will be equal to the whole-core decay power.

If the whole-core power is ORIGEN, the sum of class powers including the effects of sensitivity coefficients 3210 and 3211 will be renormalized to the sum of class powers before these sensitivity coefficients are considered.

(1) CLSNRM
Class Normalization Flag.

(a) YES
Normalize class decay heats to whole-core decay heat.

(b) NO
No normalization.

(type = integer / character*3, default = YES)

Example

```
DCH_NRM NO
```

2.1.4 Activity Calculations (BONUS)

DCH_ACT – Activity calculation options

Optional

This record specifies the switches that allow user to enable or disable activity calculations and set the format of input. If RN package is not active, activity calculations can not be enabled.

(1) IACTIV

Option specifying if activity calculations will be performed or not:

(a) ON or 1,
the activity calculations will be performed in MELCOR.

(b) OFF or 0,
the activity calculations will not be performed in MELCOR.

(type = character * 3/integer, default = 'OFF')

The following field is required if and only if IACTIV = 'ON' or '1':

(2) INPFORM

Option specifying the format of user input for decay heats and activities calculations:

(c) OLD or 0,
the old DCH input is to be used (containing time-decay heats tables for elements). DCH_SUR record can be used to define the necessary output data.

(d) NEW or 1,
the activity and decay heats calculations will be performed within the code using initial masses pre-calculated by BONUS or some other code. The user

must set only classes (DCH_CL record) that will be considered in the problem with the list of elements and the filename with initial inventory data (DCH_RCT record). DCH_SUR record can be used to define the necessary output data. In this case all other records are not allowed for input.

(type = character *3/integer, default = 'OLD')

Example

```
DCH_ACT          ON       NEW
```

DCH_SUR Output activity data

Optional. Allowed if IACTIV = ON on DCH_ACT record

This record specifies the control volumes or heat structures surfaces for which isotope activities data will be printed to the output. If the specified type is control volume, the data on isotopes activities will be summarized for the control volume and heat structure surfaces associated with the corresponding control volume. By default isotopes activities will be printed for flow paths set on FL_MACCS record and different control volume types.
Optional. Allowed if IACTIV = ON on DCH_ACT record

(1) NUMACT

Number of control volumes and surfaces for isotopes activities output.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMSUR:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) TYPE

Key specifying either heat structure surface or control volume.

(a) 1 or CVH, Control volume
(b) 2 or LHS, Left side of heat structure
(c) 3 or RHS, Right side of heat structure

(type = integer / character*3, default = none, units = none)

(3) NAME

Heat structure name defined in the HS input or control volume name defined in the CVH input depending on IKEY value.

(type = character*16, default = none, units = none)

(4) IKEY

Key specifying either data should be printed for all isotopes or for a list of isotopes set by the user.

(a) -1 or ALL, All isotopes
(b) 1 or ISOTOPE, Particular isotopes.

(type = character*7, default = none, units = none)

The following data must be set if and only if IKEY = "ISOTOPE' or "1"

(5) Isotope1

Isotope name (must coincide with name specified in database file containing isotopes physical properties in form of "Xy-nnn", where Xy – chemical element name ("Cs", "Mo", etc.) and nnn – mass number). Notice, that the element name should be written as in chemical formulas in capital first letter and small second letter.

(type = character*7, default = none, units = none)

……..

(5+NUMIS-1) IsotopeN Isotope name.

Example

```
DCH_SUR   2   ! Number   Type                    Name          IKEY      Isotopes
                    1     LHS         'HS12201-UPPER HE'      ALL
                    2     CVH            'CV530-rx bldg'      ISOTOPE    Cs-137    I-132
```

2.2 DCH Example Input for Activity Calculation

```
DCH_INPUT ! DCH package start record
DCH_ACT   ON   NEW
DCH_RCT  PWR "MyInventory.txt" "REACTOR_PWR"
DCH_CL    'XE'     USER   1   !      N    Element name
1    'XE'
!-------------------------------------------------------
DCH_INPUT ! DCH package start record
DCH_ACT ON OLD
DCH_SUR 2  ! N TYPE         NAME        IKEY
             1  CVH  CV901-ENVIRONMEN    ALL
             2  LHS  HS4011-135N-FLOO ISOTOPE  "Xe-131" "Cs-137"
```

2.3 MELCOR Input

There is at present no MELCOR input to the decay heat package.

2.4 Additional Library Input Files (BONUS)

2.4.1 File 'FissProd.in' (properties of fission products)

The data set in file "FissProd.in" by default are prepared based on tables England 1994 T.R.England, B.F.Rider. Evaluation and Compilation of Fission Yields, ENDF-349, LA-UR-94-3106; http://ie.lbl.gov/toi.html, http://ie.lbl.gov/fission.html

The data are presented in the table form, each line corresponding to one nuclide. The first line is the table header.

Column #	Notation	Meaning
1	A	mass number
2	Name	nuclide name[1]
3	$t_{1/2}$	half-life period[2]
4	E_gamma	gamma decay energy, MeV
5	E_beta	beta decay energy, MeV
6	E_alpha	alpha decay energy, MeV
7	sig_c_b	cross section of the radiation capture of thermal neutron, barn
8	Ind-235%	independent yield for ^{235}U thermal neutron induced fission, %
9	Cum-235%	cumulative yield for ^{235}U thermal neutron induced fission, %
10	Ind-239%	independent yield for ^{239}Pu thermal neutron induced fission, %
11	Cum-239%	cumulative yield for ^{239}Pu thermal neutron induced fission, %
12	Ind-241%	independent yield for ^{241}Pu thermal neutron induced fission, %
13	Cum-241%	cumulative yield for ^{241}Pu thermal neutron induced fission, %

The data format corresponds to the FORTRAN format '(I5, 5X, A3, 2X, A9, 1X, 8F10.1)'. The energy and the cross-section fields can be empty; in this case the data are interpreted as zeros.

2.4.2 File 'FPChains.in' (structure of isobaric chains)

The isobaric β^-- and γ-decays of FPs, which do not change the mass number, are taken into account in BONUS. It is supposed that the β^--decay results in one or two daughter nuclides (in the ground and isomeric states). Thermal neutron capture is also taken into account, which also produces one or two daughter nuclides with a mass number increased by 1 and therefore results in coupling of the current chain with the next one(s).

The data on the chains are arranged in the following order. The first and the second lines contain the headers while the following lines describe the chain composition and branching ratio(s). The data for the different isobaric chains are separated by a string with arbitrary

[1]Standard element name, possibly followed by a symbol '*' to denote the isomeric state.
[2] This a string 'stable' for the stable nuclides. For the radionuclides the string contains the number in scientific notation (e.g 0.13, 1.7e10) followed by a unit ('s', 'm', 'h', 'd' or 'y' for the second, minute, hour, day or year, respectively), the blank separator(s) being allowed between the number and the unit.

(> 0) number of symbols '-'. The placement of the chains in the file is arbitrary, though the coupled chains should follow each other.

For each chain, the first line contains the mass number, formatted '(I5)'. This line is followed by *n* lines, where *n* is the chain length (number of nuclides in the chain), where each line contains 7 fields and is formatted '(5A5,2f10.0)':

field #	format	meaning
1	'(A5)'	nuclide name (case sensitive)
2	'(A5)'	the name of the first daughter nuclide with the same mass number
3	'(A5)'	the name of the second daughter nuclide with the same mass number
4	'(A5)'	the name of the first daughter nuclide with the next mass number
5	'(A5)'	the name of the second daughter nuclide with the next mass number
6	'(f10.0)'	branching coefficient for the first pair of the daughter nuclides
7	'(f10.0)'	branching coefficient for the second pair of the daughter nuclides

The nuclide names in the file (field 1) should correspond to the names from the previously described file 'FissProd.in', otherwise the execution is terminated with the diagnostics 'ERROR: Nuclide xxx - mmm is defined in FPChains.in file, but is not defined in database (FissProd.in file)' , where 'xxx' is the name and 'mmm' is the mass number of the unidentified nuclide.

The first pair of daughter nuclides corresponds to β- or γ-decay. If there are two nuclides, field #6 contains the branching coefficient (the number between 0 and 1) for the first daughter (indicated if field #2). For the sole daughter nuclide, fields 3 and 6 are empty. Similarly, the second pair of daughter nuclides corresponds to (n,γ)-capture. If the capture produces two nuclides, field #7 contains the branching coefficient for the first daughter (indicated if field #4). For the sole daughter nuclide, fields 5 and 6 are empty.

2.4.3 File ' InitInventory.in' (structure of initial inventory file)

This file contains masses of isotopes at the moment of reactor shutdown. Since data can be provided for different reactor campaigns, each block should start with REACTOR_XYZ line, where XYZ… is user-defined characters distinguishing the campaign name (for example, REACTOR_PWR1Y, REACTOR_BWR_2Y). The block name used in particular calculations is set on **DCH_RCT** record.

The subsequent lines following the line containing the reactor campaign name, should define isotope names and masses in kilograms with each line containing two fields:

field #	format	meaning
1	'(A5)'	nuclide name (case sensitive)
2	'(f10)'	nuclide mass in kg

The names in the file (field 1) should correspond to the names from the previously described file 'FPChains.in', otherwise the execution is terminated with the diagnostics 'ERROR: nuclide name: xxx-mmm is not set in FPChains.in file, describing structure of isobaric chains', where 'xxx' is the name and 'mmm' is the mass number of the undefined nuclide.

Example:

```
REACTOR_PWR    ! Masses of isotopes in Kg

Se-80  4.06405E-01

Se-81  1.31432E-05

Br-81  6.81138E-01

Se-82  1.04052E+00

Se-83  1.77636E-05

Se*-83 7.74357E-07

Br-83  2.20323E-04

Kr*-83 1.67048E-04

Kr-83  1.63820E+00

Se-84  8.22949E-06

Br-84  8.15545E-05

Br*-84 7.97802E-07

Kr-84  3.07100E+00
```

```
Br-85  9.81070E-06

Kr*-85 8.35679E-04

Kr-85  1.08368E+00

Rb-85  2.81311E+00

Br-86  3.54034E-06
```

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. Their use is described in Section 7 of the MELCOR Executive (EXEC) package Users' Guide.

3.1 DCH Sensitivity Coefficients

The sensitivity coefficients for the DCH package have identifier numbers from 3200 through 3299.

3200 – Multiplier for ANS Decay Heat Curve

This sensitivity coefficient is the multiplier M_{user} given in the ANS decay heat equation (see Section 3.2 of the DCH Package Reference Manual).

(1) Multiplier for ANS decay heat curve.
(default = 1.0, units = none, equiv = ANSMLD)

3201 – Energy per Fission for Nuclides

These sensitivity coefficients are the Q factors in the ANS decay heat equation (see Section 3.2 of the DCH Package Reference Manual). The defaults are computed from SANDIA-ORIGEN data giving the fissioning rates for each nuclide. The energies include both fission and decay energies.

(1) Energy per thermal fission of U-235.
(default = 199.0 units = MeV/fission, equiv = none)

(2) Energy per thermal fission of Pu-239.
(default = 210.2, units = MeV/fission, equiv = none)

(3) Energy per fast fission of U-238.
(default = 199.3, units = MeV/fission, equiv = none)

3202 – Times for ANS Decay Heat Power Tables

These sensitivity coefficients provide the times at which the ANS decay heat powers for the nuclides and the neutron capture correction factors are available. The defaults, taken from the 56 tabulated times in the ANS standard (see Section 3.2 of the DCH Package Reference Manual), run from 0.0 s to 10^9 s.

(1–56) Times after shutdown TIMDCH(I) for decay heat tables (sensitivity coefficient 3203) and neutron capture factors (sensitivity coefficient 3204).

(default = tabulated times from ANS standard, units = s, equiv = none)

3203 – Decay Heat Powers for ANS Decay Heat Curve

These sensitivity coefficients are the factors $F(t,\infty)$ needed to compute *F(t,T)* used in the ANS decay heat equation (see Section 3.2 of the DCH Package Reference Manual). The calculation is $F(t,T) = F(t,\infty) - F(t+T,\infty)$.

(1–56, 1–3) Decay heat powers DCHPOW (I, J), corresponding to times TIMDCH(I) in sensitivity coefficient 3202, resulting from decay of products fissioning of nuclide J, where J indicates nuclide:

J = 1, U-235
J = 2, Pu-239
J = 3, U-238

(default = values from Tables 4, 5, and 6 of the ANS standard, units = MeV/fission, equiv = none)

3204 – Neutron Capture Correction factors for ANS Decay Heat Curve

These sensitivity coefficients are the factors $G_{max}(t)$ that are used for *G(t)* for shutdown times greater than 10^4 seconds. *G(t)* is used in the ANS decay heat equation (see Section 3.2 of the DCH Package Reference Manual).

(1–56) Neutron capture correction factors CAPNEU(I), corresponding to times TIMDCH(I) in sensitivity coefficient 3202.

(default = values from Table 10 of the ANS standard, units = none, equiv = none)

3205 – Parameters for Actinide Decay Heat Calculation

These sensitivity coefficients are the parameters in the ANS decay heat equation for the decay power from U-239 and Np-239 (see Section 3.2 of the DCH Package Reference Manual). The default for the first parameter is computed so that Equation 16 of the ANS standard matches the ORIGEN results for decay power from U-239 and Np-239. The other defaults are taken from the discussion of the equation on page 5 of the ANS standard.

(1) Number of atoms of U-239 produced per second per fission per second at time of shutdown.
(default = 0.526, units = 1/fission, equiv = ATU239)

(2) Average energy from decay of U-239 atom.
(default = 0.474, units = MeV, equiv = EU239)

(3) Average energy from decay of Np-239 atom.
(default = 0.419, units = MeV, equiv = ENP239)

(4) Decay constant for U-239.
(default = 4.91E-4, units = s^{-1}, equiv = DU239)

(5) Decay constant for Np-239.
(default = 3.41E-6, units = s^{-1}, equiv = DNP239)

3210 – Multiplier for all ORIGEN Elemental Decay Heat Curves

This sensitivity coefficient is a multiplier that will be applied to all elemental decay heat power curves stored as default data in MELCOR.

(1) Multiplier for ORIGEN elemental decay heat power.
(default = 1.0, units = none, equiv = ORGALL)

3211 – Multipliers for Individual ORIGEN Elemental Decay Heat Curves

These sensitivity coefficients are multipliers for individual elemental decay heat power curves stored as default data in MELCOR. These multipliers will be applied in addition to the general multiplier ORGMUL. The list of relevant elements is core-model specific; the list given here is for the LWR COR package. Users of an alternate core package should consult the documentation for that package.

(1-29) Multiplier for individual elemental decay power. The indices for each element are:

1.	AS	2.	SE	3.	BR	4.	KR	5.	RB	6.	SR	7.	Y
8.	ZR	9.	NB	10.	MO	11.	TC	12.	RU	13.	RH	14.	PD

15.	AG	16.	SN	17.	SB	18.	TE	19.	I	20.	XE	21.	CS
22.	BA	23.	LA	24.	CE	25.	PR	26.	ND	27.	PM	28.	U
29.	NP												

(default = 1.0, units = none, equiv = none)

3212 – Fraction of Fuel Equilibrium Cycle Elapsed

This sensitivity coefficient, which must have a value between 0.0 and 1.0, is used in determining the fission product mass inventory; 0.0 corresponds to the beginning of the fuel cycle (1/3 fresh fuel, 1/3 with one year in core, 1/3 with two years in core), and 1.0 corresponds to the end of the fuel cycle (1/3 of fuel with one year in core, 1/3 with two years in core, 1/3 with three years in core). These data are used only if WCTYPE on card DCHDECPOW is ORIGEN.

(1) Fraction of fuel equilibrium cycle that has elapsed at beginning of problem.

(default = 1.0, units = none, equiv = TFLCYC)

DCH_SC – DCH Sensitivity Coefficients Input Record

Optional

This table of records provides the means for the user to change default sensitivity coefficients for the DCH package.

(1) NUMSTR

Number of table records.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the DCH sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

First sensitivity coefficient index.

(type = integer, default = none, units = none)

(5) NB

Second sensitivity coefficient index. Specific sensitivity coefficient dependent.

(type = integer, default = none, units = none)

Example

```
DCH_SC   4  ! N   SC          VALUE    INDEX1  INDEX2
              1   3201        200.0         3
              2   3203        5.0          16       1
              3   3203        8.8           6       2
              4   3203        0.05         56       3
```

4 Plot Variables

The plot variables are described in this section. Note that control function arguments (some are identical in definition to these plot variables but different in format) are described in the following section.

DCH-COREPOW.0	Whole-core decay heat power. (units = W)
DCH-CLSPOW.n	Decay heat power per unit mass for radionuclide class n. (units = W/kg)
DCH-TOTCLSPOW.0	Total decay heat power for all radionuclide classes. (units = W)

5 Control Function Arguments

The control function arguments are described in this section. Note that plot variables (some are identical in definition to these control function arguments but different in format) are described in the previous section.

DCH-COREPOW	Whole-core decay heat power. (units = W)

DCH-CLSPOW(NameCLS)	Decay heat power per unit mass for radionuclide class NameCLS. (units = W/kg)
DCH-TOTCLSPOW	Total decay heat power for all radionuclide classes. (units = W)

6 Example Input

The following example input to MELCOR will input the default decay heat data set in MELCOR. The input will be in three sections: general, whole-core, and radionuclide class.

6.1 General Input

```
! REQUEST PWR DEFAULT REACTOR POWER AND

! RADIONUCLIDE INVENTORIES
DCH_RCT PWR

! SET REACTOR SHUTDOWN TIME TO DEFAULT – 0.0 S, NO SHUTDOWN FUNCTION.
DCH_SHT TIME 0.0
```

6.2 Whole-Core Input

```
! SELECT ANS DECAY CURVE FOR WHOLE-CORE DECAY HEAT
DCH_DPW ANS

! SET REACTOR OPERATING TIME TO DEFAULT – 0.8*2 YEARS
DCH_OPT 5.05E7

! SET TOTAL REACTOR FISSION POWER DUE TO:
! U-235 2208.6 MW (DEFAULT FOR PWR)
! PU-239 1059.8 MW (DEFAULT FOR PWR)
! U-238 143.6 MW (DEFAULT FOR PWR)
DCH_FPW 2.2086E9 1.0598E9 1.436E8

! SET PSI IN NEUTRON CAPTURE CORRECTION EQUATION TO DEFAULT
DCH_NFA 0.713
```

6.3 Radionuclide Class Input

```
! DEFINE NEW ELEMENTS
! NEW ELEMENT XX

!     NAME MASS(KG) Decay heat table len
DCH_EL XX  0.100      2
! N TIME (S) DECAY HEAT (W/KG)
  1 0.0 10.0
  2 5.0 0.01

! NEW ELMENT YY
!     NAME MASS(KG) Decay heat table len
DCH_EL YY  9.900    3
1 0.0   0.1
2 10.0  0.01
```

```
3 100.0 0.001

! ********** DEFINE CLASSES **********
! SELECT ALL DEFAULT CLASSES
DCH_CL 'XE' DEFAULT
DCH_CL 'CS' DEFAULT
DCH_CL 'BA' DEFAULT
DCH_CL 'I2' DEFAULT
DCH_CL 'TE' DEFAULT
DCH_CL 'RU' DEFAULT
DCH_CL 'FE' DEFAULT
DCH_CL 'CE' DEFAULT
DCH_CL 'LA' DEFAULT
DCH_CL 'UO2' DEFAULT
DCH_CL 'CD' DEFAULT
DCH_CL 'AG' DEFAULT
DCH_CL 'BO2' DEFAULT
DCH_CL 'H2O' DEFAULT
DCH_CL 'CON' DEFAULT
DCH_CL 'CSI' DEFAULT
! DEFINE CLASS 17 – NEW ELEMENT XX
DCH_CL 'FIRST NEW CLASS' USER 1
       1 XX

! DEFINE CLASS 18 – NEW ELEMENTS YY AND XX
DCH_CL 'SECOND NEW CLASS' USER 2
     1 YY
     2 XX

! DEFINE CLASS 19 – NEW ELEMENT YY + DEFAULT ELEMENT PM
DCH_CL 'THIRD NEW CLASS' USER 2
     1 YY
     2 PM
```

6.4 Sensitivity Coefficients Input

```
DCH_SC   17  ! N   SC           VALUE    INDEX1  INDEX2
! ANS DECAY HEAT SENSITIVITY COEFFICIENTS
! SET ANS MULTIPLIER TO DEFAULT
                1   3200           1.00     1
! SET ENERGY PER FISSION FOR U-235 DEFAULT – 199 MEV/FISSION
                2   3201           199.0    1
! SET ENERGY PER FISSION FOR PU-239 DEFAULT – 210.2 MEV/FISSION
                3   3201           210.2    2
! SET ENERGY PER FISSION FOR U-238 DEFAULT – 199.3 MEV/FISSION
                4   3201           199.3    3
! SET TIME AT WHICH DECAY HEAT POWER AND NEUTRON
! CAPTURE CORRECTION FACTOR GIVEN – FIRST POINT ONLY
! TIME = 0.0 S AFTER SHUTDOWN
                5   3202           0.0      1
! SET DECAY HEAT POWER FOR U-235 AT FIRST TIME POINT –
! DEFAULT = 13.18 MEV/FISSION
                6   3203           13.18    1     1
! SET DECAY HEAT POWER FOR PU-239 AT FIRST TIME POINT –
! DEFAULT = 10.93 MEV/FISSION
                7   3203           10.93    1     2
! SET DECAY HEAT POWER FOR U-238 AT FIRST TIME POINT –
! DEFAULT = 16.23 MEV/FISSION
                8   3203           16.23    1     3
! SET NEUTRON CAPTURE CORRECTION FACTOR GMAX AT FIRST TIME POINT
! DEFAULT = 1.02
                9   3204            1.02    1
```

```
! SET PARAMETERS FOR ACTINIDE DECAY HEATS

! ATOMS OF U-239 PRODUCED PER SECOND PER FISSION PER SECOND, DEFAULT
               10   3205           0.54      1
! ENERGY FROM DECAY OF U-239 ATOM – MEV
               11   3205           0.474     2
! ENERGY FROM DECAY OF NP-239 ATOM – MEV
               12   3205           0.419     3
! DECAY CONSTANT FOR U-239 – INVERSE SECONDS
               13   3205           4.91E-4   4
! DECAY CONSTANT FOR NP-239 – INVERSE SECONDS
               14   3205           3.41E-6   5


! SET ORIGEN SENSITIVITY COEFFICIENTS

! SET MULTIPLIER FOR ALL ORIGEN DATA TO 1.0
               15   3210           1.0       1
! SET MULTIPLIER FOR BROMINE TO 1.0
               16   3211           1.0       3

! SET FRACTION OF FUEL EQUILIBRIUM ELAPSED – DEFAULT = END OF CYCLE

               17   3212           1.0       1
```

The fact that XX, YY, and PM each appear in two classes (PM is also in default class 9) does not mean that their decay heat will be double counted. Any mass that appears in any of the classes as a result of initialization or calculations in the RN package will have the specific power (W/kg) of a mixture of the elements it contains in proportion to their inventories at shutdown. Thus, any mass in class name FIRST NEW CLASS will have the specific power of element XX, while mass in class name SECOND NEW CLASS will have a specific power corresponding to a mixture of 0.1/(0.1+9.9) parts by mass of XX and 9.9/(0.1+9.9) parts of YY. Finally, any mass in class THIRD NEW CLASS will be treated as a mixture of YY and PM, in the proportions of 9.9 kg of YY to 9.8948 kg (the initial inventory) of PM. If the RN package calculates these masses in a way that conserves XX, YY, and PM, no double counting will occur.

7 Example Output

7.1 MELGEN Output

7.1.1 Standard DCH Output

Tables for user-defined elements follow.

```
TOTAL NUMBER OF DECAY HEAT F.P. CLASSES  =    18
TOTAL NUMBER OF DECAY HEAT ELEMENTS      =   105
TOTAL NUMBER OF DECAY HEAT CLASS TIMES   =   572

 EDIT OF ANS DECAY HEAT DATA
   TYPE OF REACTOR = PWR
   REACTOR OPERATING PERIOD =  6.76000E+06 SECONDS
   REACTOR SHUTDOWN TIME    =  1.00000E+20 SECONDS
   NUMBER OF FISSIONING NUCLIDES =   3
   NUMBER OF TIMES AT WHICH ANS DATA AVAILABLE =  56
   OPERATING POWER FROM FISSIONING OF U235  =  1.94191E+09 WATTS
```

```
    OPERATING POWER FROM FISSIONING OF PU239 =  9.31830E+08 WATTS
    OPERATING POWER FROM FISSIONING OF U238  =  1.26260E+08 WATTS
    TOTAL OPERATING POWER =  3.00000E+09 WATTS

 TOTAL REACTOR DECAY HEAT POWER COMPUTED FROM TABULAR FUNCTION TF50

    ** EDIT OF   18 FISSION PRODUCT CLASSES **

         *     3 USER-DEFINED CLASSES *

EDIT OF DECAY HEAT FISSION PRODUCT CLASS - CSI
   CLASS MASS =  1.00000E-06 KG
   NUMBER OF ELEMENTS IN CLASS =     1
   ELEMENTS IN CLASS =
                        CI

    TIME (S)          DECAY HEAT POWER (W/KG)
    0.00000E+00             9.13500E+05
    1.00000E+00             8.54300E+05
    1.00000E+01             6.65900E+05
    6.00000E+01             4.86100E+05
    3.00000E+02             3.44900E+05
    6.00000E+02             2.97100E+05
    1.80000E+03             2.22900E+05
    3.60000E+03             1.93700E+05
    7.20000E+03             1.59500E+05
    1.08000E+04             1.36300E+05
    2.16000E+04             1.05400E+05
    4.32000E+04             8.50400E+04
    8.64000E+04             6.49400E+04
    1.72800E+05             4.63900E+04
    2.59200E+05             3.60700E+04
    6.04800E+05             1.60900E+04
    9.50400E+05             7.98500E+03

EDIT OF DECAY HEAT FISSION PRODUCT CLASS - IM
   CLASS MASS =  9.00000E-01 KG
   NUMBER OF ELEMENTS IN CLASS =     1
   ELEMENTS IN CLASS =
                        IM

    TIME (S)          DECAY HEAT POWER (W/KG)
    0.00000E+00             6.60000E+03
    1.00000E+00             6.10000E+03
    1.00000E+01             4.80000E+03
    6.00000E+01             3.50000E+03
    3.00000E+02             2.40000E+03
    6.00000E+02             2.10000E+03
    1.80000E+03             1.60000E+03
    3.60000E+03             1.40000E+03
    7.20000E+03             1.30000E+03
    1.08000E+04             1.30000E+03
    2.16000E+04             1.30000E+03
    4.32000E+04             1.20000E+03
    8.64000E+04             1.00000E+03
    1.72800E+05             8.00000E+02
    2.59200E+05             6.40000E+02
    6.04800E+05             3.30000E+02
    9.50400E+05             2.20000E+02

EDIT OF DECAY HEAT FISSION PRODUCT CLASS - HBO2
   CLASS MASS =  2.30000E-01 KG
   NUMBER OF ELEMENTS IN CLASS =     1
```

```
   ELEMENTS IN CLASS =
                        HB

    TIME (S)          DECAY HEAT POWER (W/KG)
    0.00000E+00             5.60000E+03
    1.00000E+00             5.10000E+03
    1.00000E+01             3.80000E+03
    6.00000E+01             2.50000E+03
    3.00000E+02             1.40000E+03
    6.00000E+02             1.10000E+03
    1.80000E+03             1.00000E+03
    3.60000E+03             1.00000E+03
    7.20000E+03             1.00000E+03
    1.08000E+04             1.00000E+03
    2.16000E+04             1.00000E+03
    4.32000E+04             1.00000E+03
    8.64000E+04             1.00000E+03
    1.72800E+05             7.00000E+02
    2.59200E+05             5.40000E+02
    6.04800E+05             2.30000E+02
    9.50400E+05             1.20000E+02

         *    15 DEFAULT CLASSES *

EDIT OF DECAY HEAT FISSION PRODUCT CLASS - XE
   CLASS MASS =  5.12400E+02 KG
   NUMBER OF ELEMENTS IN CLASS =     8
   ELEMENTS IN CLASS =
                        XE
                        KR
                        RN
                        HE
                        NE
                        AR
                        H
                        N

    TIME (S)          DECAY HEAT POWER (W/KG)
    0.00000E+00             1.50003E+04
    1.00000E+00             1.40291E+04
    6.12000E+00             1.22676E+04
    1.00000E+01             1.09327E+04
    1.18800E+01             1.08216E+04
    1.80000E+01             1.04602E+04
    2.98800E+01             9.75857E+03
    6.00000E+01             7.97971E+03
    6.12000E+01             7.96812E+03
    1.18800E+02             7.41170E+03
    2.41200E+02             6.22931E+03
    3.00000E+02             5.66131E+03
    6.00000E+02             4.87685E+03
    6.12000E+02             4.86468E+03
    1.18800E+03             4.28067E+03
    1.80000E+03             3.66016E+03
    3.60000E+03             3.17919E+03
    5.40000E+03             2.81155E+03
    7.20000E+03             2.44392E+03
    1.08000E+04             2.03725E+03
    1.44000E+04             1.82516E+03
    2.16000E+04             1.40099E+03
    2.88000E+04             1.25463E+03
    3.60000E+04             1.10828E+03
    4.32000E+04             9.61929E+02
```

```
5.40000E+04            8.85124E+02
7.20000E+04            7.57117E+02
8.64000E+04            6.54711E+02
1.29600E+05            5.29376E+02
1.72800E+05            4.04041E+02
2.59200E+05            3.18566E+02
3.45600E+05            2.85373E+02
5.18400E+05            2.18989E+02
6.04800E+05            1.85797E+02
6.91200E+05            1.67102E+02
8.64000E+05            1.29712E+02
9.50400E+05            1.11017E+02
```

7.2 Standard MELCOR Output

Tables for the remaining default classes 2 through 15 follow.

```
      *   DECAY HEAT PACKAGE EDIT   *

                 POWER FROM FISSION PRODUCT CLASSES

  CLASS      CLASS          POWER/MASS        POWER
 NUMBER       NAME             W/KG             W

   16     CSI               1.04985E+06     1.04985E+00
   17     IM                7.58511E+03     6.82660E+03
   18     HBO2              6.43585E+03     1.48025E+03
    1     XE                1.72392E+04     8.83336E+06
    2     CS                5.13776E+04     1.39798E+07
    3     BA                8.86589E+04     1.78648E+07
    4     I2                1.96662E+06     4.24790E+07
    5     TE                2.48097E+05     1.10899E+07
    6     RU                3.15359E+04     6.33871E+06
    7     FE                1.18018E+05     1.78254E+07
    8     CE                2.60686E+04     1.28336E+07
    9     LA                5.42869E+05     5.89827E+07
   10     UO2               6.58386E+01     3.88909E+06
   11     CD                7.36508E+06     4.42074E+06
   12     AG                1.82238E+04     4.04569E+05
   13     BO2               0.00000E+00     0.00000E+00
   14     H2O               0.00000E+00     0.00000E+00
   15     CON               0.00000E+00     0.00000E+00

TOTAL POWER FROM FISSION PRODUCT CLASSES =  1.98950E+08 W

FISSION PRODUCT CLASS DECAY POWERS ARE BEING PROPORTIONED TO
WHOLE CORE DECAY POWER

WHOLE REACTOR CORE DECAY HEAT POWER COMPUTED FROM TABULAR FUNCTION TF50
```

7.3 MELCOR Activity Output

In the output file the following data will be printed:

- Data on isotope activities summarized for all control volumes of the same type, including heat structures associated with the corresponding control volumes. These data will be provided with the edit frequency specified by user on EXEC_TIME record in MELCOR input.

Example

```
ACTIVITY (BQ) FOR CONTROL VOLUMES OF THE SAME TYPE

ISOTOPE NAME   CVTYPE01      CVTYPE02          CVTYPE03          CVTYPE04
    Se-80  0.000000E+00    0.000000E+00     0.000000E+00      0.000000E+00
    Se-81  0.415679E-30    0.108095E-30     0.250123E-30      0.736224E-28
    Br-81  0.000000E+00    0.000000E+00     0.000000E+00      0.000000E+00
    Se-82 0.412519E-10     0.107273E-10     0.248221E-10      0.730626E-08
```

– Data on isotope activity values for those flow paths that are set for MACCS2 (FL_MACCS record).

Example

```
ACTIVITY (BQ) FOR MACCS FLOW PATHS
   ISOTOPE NAME    MACCS1        MACCS2          MACCS3            MACCS4
       Se-80     0.000000E+00  0.000000E+00   0.000000E+00    0.000000E+00
       Se-81     0.000000E+00  0.000000E+00   0.000000E+00    0.000000E+00
```

– isotopes activity values for control volumes and associated heat structures and for particular side of heat structures set by the user on DCH_SUR record.

Example

```
ACTIVITY (BQ) FOR CONTROL VOLUMES AND HEAT STRUCTURE SIDES DEFINED BY THE
USER

   LHS     INLET-FLOOR        Cs-138           Se-82               Rb-85
                      0.000000E+00        0.000000E+00        0.000000E+00
   LHS     CORWALL6 15006    Cs-138           Se-82               Rb-85
                      0.543393E-13        0.000000E+00        0.000000E+00
                            CVH              CVH
                      CORE-INLET100     CONTAIN 500
             Se-80    0.000000E+00      0.000000E+00
             Se-81    0.000000E+00      0.000000E+00


        ***************************************
         *                                     *
         *           ACTIVITY    EDIT          *
         *                                     *
         ***************************************


ACTIVITY (BQ) FOR CONTROL VOLUMES AND HEAT STRUCTURE SIDES DEFINED BY THE USER

  LHS     INLET-FLOOR         Cs-138              Se-82               Rb-85
                            0.00000E+00      0.000000E+00        0.000000E+00
  LHS     CORWALL6 15006      Cs-138              Se-82               Rb-85
                            0.543393E-13      0.000000E+00        0.000000E+00
                             CVH               CVH
                         CORE-INLET100      CONTAIN 500
             Se-80        0.000000E+00      0.000000E+00
             Se-81        0.000000E+00      0.000000E+00
             Br-81        0.000000E+00      0.000000E+00
             Se-82        0.000000E+00      0.000000E+00
```

```
          Se-83      0.000000E+00      0.000000E+00
         Se*-83      0.000000E+00      0.000000E+00
          Br-83      0.000000E+00      0.000000E+00
         Kr*-83      0.000000E+00      0.000000E+00
          Kr-83      0.000000E+00      0.000000E+00

ACTIVITY (BQ) FOR CONTROL VOLUMES OF THE SAME TYPE

     ISOTOPE NAME   CVTYPE01          CVTYPE02          CVTYPE03
          Se-80      0.000000E+00      0.000000E+00      0.000000E+00
          Se-81      0.415679E-30      0.108095E-30      0.250123E-30
          Br-81      0.000000E+00      0.000000E+00      0.000000E+00
          Se-82      0.412519E-10      0.107273E-10      0.248221E-10
  Se-83   0.265134E-23   0.689466E-24   0.159536E-23
```

8 Diagnostics and Error Messages

An error message is printed if the DCH package is asked to compute the decay heat for an invalid class name. The class number and calling package name are printed.

An error message is printed if the problem time is off the range of the decay heat table.

External Data File (EDF) Definition Record Package Users' Guide

The External Data File (EDF) Definition Record package serves as a utility in MELCOR to allow a general means of communication with external data files containing time history data. In any MELCOR run, one or more such files may be defined. Each file contains values of time and the corresponding values of one or more dependent variables and, therefore, defines each dependent variable as a function of time. Data can be read from or written to each file; the permitted direction is defined in MELGEN. All characteristics of each file are also defined by EDF input. The EDF package assumes all responsibility for opening and positioning the files, and for reading from each file or writing to it, as appropriate.

Data that have been read in by the EDF package may be accessed and used by other MELCOR packages. Data to be written out are obtained from the databases of other MELCOR packages. The interface may be either through the Control Function (CF) package or—in cases where appropriate coding has been provided—through utility-level entries in the EDF package.

The primary use of the EDF package is to facilitate input of data that define sources and/or boundary conditions as a function of time, particularly in cases where the volume of data is so great that the generation of tabular function input would be extremely tedious and error prone, or completely impractical. The ability to write files allows these input data to be generated directly by other MELCOR calculations; files written by other codes (or even manually) may also be used. In addition, data files written by MELCOR could be used as input to other simulation codes or sent to an output processor.

This document introduces the EDF package; describes input requirements; lists sensitivity coefficients, plot variables, and control function arguments; and describes output. Sample input is provided and discussed.

Contents

1 INTRODUCTION 3

2 USER INPUT REQUIREMENTS 4

2.1 MELGEN User Input 4

EDF_ID – External Data File Definition Record 4
EDF_CHN – Number of Channels and Channel Variables for WRITE or READ File 5
EDF_FMT – External Data File Format 6
EDF_TIM – Time Offset for External Data File 6
EDF_DTW – Write Increment Control for WRITE or PUSH File 7

2.2 MELCOR User Input 8

EDF_ID – External Data File Definition Record 8
EDF_FMT – External Data File Format 8
EDF_TIM – Time Offset for External Data File 8
EDF_DTW – Write Increment Control for WRITE or PUSH File 8

3 SENSITIVITY COEFFICIENTS 9

4 PLOT VARIABLES 9

5 CONTROL FUNCTIONS ARGUMENTS 9

6 EXAMPLE INPUT 9

7 DISCUSSION OF OUTPUT 11

APPENDIX 1 12

1 Introduction

The External Data File (EDF) Definition Record package serves as a utility in MELCOR to allow other packages to communicate with external data files. One or more such files may be defined in any MELCOR calculation. Each file is identified by a user-defined name by which the operating system recognizes the file and contains values of time and one or more dependent variables. It therefore defines a function of time, referred to as a data channel, for each dependent variable in the file. Data can be either read from or written to each file; the permitted direction is defined in MELGEN. A file may not be both read from and written to in a single calculation.

A file from which data are read is termed a READ file; the data that have been read may be accessed by other MELCOR packages, either as control function arguments through the Control Function (CF) package or directly through utility-level calls to the EDF package. In the former case, only old values, appropriate to the start of the current timestep, are defined. (This is the case for <u>any</u> control function value.) In the latter case, only values for times within the current timestep are available because only that portion of the external file is required to be in the EDF database. Simple linear interpolation is used between tabulated points read from the file. Direct access requires specific coding that must be provided by the code developer. This capability is currently used only by the Control Volume Hydrodynamics (CVH) and Transfer Process (TP) packages.

Data to be written are obtained from other MELCOR package databases. Time is accessed directly and automatically by EDF; dependent variables may be obtained in either of two ways. For a WRITE file, the value of any variable accessible as a control function argument may be pulled into the EDF database using methods that exactly parallel those used for arguments in the CF package. For a PUSH file, values must be pushed into the EDF database by another package through utility-level entries in the EDF package. Such direct access again requires specific coding in the controlling routine, and the option is currently used only by the TP package. The frequency with which records are written is controlled by the user.

With the current EDF coding, the variable values that are to be written to a file are also available as control function arguments. However, use of these control function arguments should be avoided because the values available in EDF at the time the CF package is called are one timestep out of date. (If the order of execution were reversed, the values of any control functions written to an external file would be out of date.) In essentially all cases, up-to-date information may be obtained by direct reference to a control function argument in the package from which EDF obtains the data.

If data are pushed into the EDF database by other packages, EDF has no way of identifying (for editing purposes) what the data represent. Therefore, a path is provided to allow other packages to also define character labels for the data channels pushed into EDF. A check is made in the EDF package to verify that values are actually pushed for every channel on every timestep.

The primary use of the EDF package is to facilitate input of data that define sources and/or boundary conditions as functions of time. The most obvious use is to communicate data from one MELCOR run to another. For example, ex-vessel sources can be calculated with a detailed nodalization of the primary system and a very simple containment model. These sources can be stored in a file and then input to a later calculation with detailed containment nodalization (but with little or no representation of the primary).

In cases where an external data file specifies a source or sink of mass or energy, it is generally preferable that the file record cumulative (integral) sources rather than rates. This assures that, regardless of the actual timesteps taken by MELCOR, the total source in the calculation will match that in the data file. Construction of an integral source may involve use of the INTEGRAL control function in the MELCOR run which writes the file. If the receiving package does not accept cumulative sources, data read from this file in a later run can be converted to rates using the forward difference (DER-F) control function.

The EDF package can also be used to communicate data to or from other codes. An input file need not be generated by MELCOR; it can be the output of another code, or could even be constructed manually. A file written by MELCOR can be used as input to another simulation code or to an output processor, such as a special-purpose plot program. To facilitate these uses, the format of records in each file may be specified in MELGEN; the default is for the file to be unformatted.

EDF files have also been used to generate tabular data, in prescribed standard formats, for code comparison exercises. In such cases, the required files can be written directly by MELCOR in the specified format, with no code modifications required, if all desired quantities are available (or can be constructed) as control function arguments.

On each restart, including the initial MELCOR execution from the restart file generated by MELGEN, a limited check is made of the data in each file to be read. It is required that the last record before the current time, match a record read during the previous execution; a warning is issued if the position of this record in the file has been altered. The intent is to detect obvious errors without preventing the user from editing the file, if desired.

2 User Input Requirements

A full description of all external data files must be provided as part of MELGEN input. Certain elements of the description may be changed on restart through input to MELCOR.

2.1 MELGEN User Input

The user input for the EDF package is described below. One set of records is required for each external data file; the actual records required differ for different types of files.

EDF_ID – External Data File Definition Record

Required

This record defines the user-defined name for the data file, the direction and mode of information transfer, and the name by which the file is known to the operating system. It is the character string that will be used in the FILE-FILNAM parameter when the file is opened. If FILNAM contains lower case characters that must be preserved, enclose it in single quotes (').

(1) EDFNAM

User defined external data filename.

(character*16, default = none, units = none)

(2) MODE

Direction and mode of information transfer.

(a) 0 or READ
(b) 1 or WRITE
(c) 2 or PUSH

(type = integer/character*5, default = none)

(3) FILNAM

Name of file recognized by the operating system, such as 'input47.dat'.

(type = character*80, default = none)

Examples:

```
!      EDFNAM MODE   FILNAME
EDF_ID EDF1   WRITE  EDF_SRVDATA
EDF_ID EDF2   WRITE  EDF_CVHPAR.TXT
EDF_ID EDF3   1      EDF_RESULTS.TXT
EDF_ID EDF4   PUSH   EDF_MELTEJ.DAT
```

EDF_CHN – Number of Channels and Channel Variables for WRITE or READ File

Required

(1) NCHAN

Number of channels (dependent variables) in each record of the file.

(type = integer, default = none, units = none)

Next fields are input for WRITE file only as a list with length NCHAN.

(1) NARG

Argument index.

(type = integer, default = none)

(2) CHARG

Control function argument to identify an element of the database.

(type = character*24, default = none)

Values for each channel (dependent variable) of a WRITE data file are obtained by reference to control function arguments. The variables available in this way are listed in the users' guides for the various packages. Exact NCHAN arguments are required; they are assigned to channels in the order they are input in the table. Note that the independent variable, TIME, is automatically written as the first datum in each record of the EDF file and should not to be input in the table.

Examples:

```
! for PUSH file
EDF_ID   EDF1  PUSH  EDF_MELTEJ.DAT
EDF_CHN  16    ! number of channels
! for WRITE file
EDF_ID   EDF1  WRITE EDF_SRVDATA
EDF_CHN  7 !N Value Name
            1 CVH-P(CV150)
            2 CVH-P(CV200)
            3 FL-VEL(FL199,P)
            4 FL-VEL(FL199,A)
            5 FL-MFLOW(FL199, FOG)
            6 FL-MFLOW(FL199, H2O-VAP)
            7 FL-MFLOW(FL199, H2)
```

EDF_FMT – External Data File Format

Optional

A format may be defined for each external data file; the default is for the file to be unformatted. The field specifier for TIME in WRITE or PUSH files should provide sufficient significant digits to distinguish the written time values. The format may specify that a logical record contain more than one physical record (line in the file). A physical record length in excess of 133 characters may cause problems on many systems.

(1) IFMT

Format of records in the external file. If the record is omitted or blank, the file is assumed to be unformatted. If the format contains one or more commas, it must be enclosed in single quotes. The enclosing parentheses may be included '(4E12.4)' or omitted '4E12.4'; in the latter case, the input string is limited to character*22.

(type = character*24, default = " ")

Examples

```
EDF_FMT '(1P,8E12.5)'
EDF_FMT '(33(1X,E11.4))'
EDF_FMT '(17E15.7)'
EDF_FMT '(41(1X,E11.4))'
```

EDF_TIM – Time Offset for External Data File

Optional

To simplify communications between different codes, the zero of time in a data file need not coincide with that in the MELCOR run. The relationship is tfile = tMELCOR + toff. The offset may be positive or negative.

(1) TIMOFF

Offset of time in the external data file relative to time in MELCOR.

(type = real, default = 0.0, units = s)

Example

```
EDF_TIM   5.0
```

EDF_DTW – Write Increment Control for WRITE or PUSH File

Required for WRITE or PUSH File

The output of records to a WRITE or PUSH file is controlled by the user. The control consists of (start time, time increment) pairs. It is similar to that for MELCOR output as described in the MELCOR/MELGEN Executive (EXEC) Users' Guide, but no output is generated until the first time value has been passed. At least one such pair must be input.

(1) LENTAB

Length of time table.

(type = integer, default = none, units = none)

The following data are input as a table with length LENTAB:

(1) NT

Table record index.

(type = integer, default = none, units = none)

(2) TWEDF

Time at which the following output increment takes effect.

(type = real, default = none, units = s)

(3) DTWEDF

Time increment between output records.

(type = real, default = none, units = s)

Example

```
EDF_DTW   2 !NT     TWEDF       DTWEDF
            1        0.0        50000.0
            2     200000.0      5000.0
```

2.2 MELCOR User Input

Certain elements of the input for an external data file may be changed at restart (this includes the first execution of MELCOR). This is intended to allow the user to replace or rename files, or to alter the frequency with which records are written to WRITE or PUSH files.

The input to MELCOR is a subset of that described for MELGEN; the permitted records are described below (the presence of any others will be ignored).

EDF_ID – External Data File Definition Record

Required if some other data are changed for the EDF file

The name for the data file, the direction, and mode of information transfer must agree with the original values input to MELGEN on the corresponding EDF_ID record. The name of the file on the operating system FILNAM may be changed. The record must be input in case some of the other data associated with this external data file will be changed on the other records when running MELCOR.

EDF_FMT – External Data File Format

Optional

The format of the file may be changed. The user should be aware that for a WRITE or PUSH file an attempt will be made to open and read the old file under the new format in order to position it correctly. This could result in a read error if the old file exists and the old and new formats are incompatible.

EDF_TIM – Time Offset for External Data File

Optional

EDF_DTW – Write Increment Control for WRITE or PUSH File

Optional

Note that on any restart, a limited check is made for changes in a READ file. An error is assumed if the last record found on the newly connected file before the time of the restart does not match any record already in the buffer (read from the file previously connected). If there is a matching record but the record number is different, a warning message is issued to the output and diagnostic files.

3 Sensitivity Coefficients

There are no sensitivity coefficients associated with the EDF package.

4 Plot Variables

The plot variable currently included in the EDF package is described below. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

EDF.n.m	Value of the mth data channel in external data file index n. If the data are plotted, the units will be given (by default) as UNK because they are not—and cannot be—known by EDF. The user may, of course, specify the correct units by defining a nondefault axis label as part of the input to the plot program.

5 Control Functions Arguments

The variables in the EDF package that may be used for control function arguments are described below. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

EDF(NameEDF,m)	Value of the mth data channel in external data file NameEDF.

The use of channels in WRITE and/or PUSH files as control function arguments should be avoided because the values available at the time the CF package is called are one timestep out of date. In most cases, up-to-date values of the underlying variables may be obtained by direct reference to control function arguments in the packages from which EDF obtains the data.

6 Example Input

An example of input for a READ file is:

```
EDF_ID   MELT-SOURCE   READ   RUN27.DAT
EDF_CHN   4
EDF_FMT   5E12.5   ! Time and four dependent variables
EDF_TIM   2600.    ! Melcor time 0 corresponds to 2600 s in file
```

This defines an external data file with the user-defined name MELT-SOURCE and will cause data, consisting of time and four dependent variables, to be read from the file RUN27.DAT under FORMAT 5E12.5. The (interpolated) values of the dependent variables

are available as control function arguments with the names EDF('MELT-SOURCE', 1) through EDF('MELT-SOURCE', 4).

The data from a READ file may always be accessed as control function arguments; in general, this is the only method available to the user. However, in a few cases specific coding has been included in the package to access EDF data directly. At present only two packages make use of this capability:

(1) The CVH package allows a property of a time-specified (boundary) volume to be specified by direct reference to the contents of an external data file. The field CFTFSELECT for a property of time-specified volume equal to EDF or two (as described in the CVH Users' Guide) will result in the value of the corresponding property being extracted directly from channel NUMCHN of external data file EDFNAM.

(2) The TP package can be instructed to actively construct parcels of debris (or of radionuclides) from data contained in a READ file specified in a TP_IN record. These parcels are treated exactly as if they had been received from another package (e.g., from Core [COR]); they are held until called for by another package (e.g., by Cavity [CAV]). This allows debris and/or radionuclides to be added to a MELCOR calculation at a rate determined by another calculation performed by MELCOR, by another code, or manually. The user has relatively few options in defining such a file. See the CAV, COR, RadioNuclide (RN), and TP Users' Guides for more information. An example of input for a WRITE file is:

```
EDF_ID   SPECIAL-DATA   WRITE   'specdat.dat'
EDF_CHN  3 !N Value Name
            1 CVH-P(CV200)
            2 CF-VALU(CF4)
            3 CAV-MEX(CAV3, H2)
EDF_DTW   3 ! NT    TWEDF  DTWEDF
               1     500.    100.
               2     1000.   10.
               3     5000.   1000.
```

This defines an external data file with the user-defined name SPECIAL-DATA and will cause unformatted records to be written to the file specdat.dat. A record will be written every 100 s starting at 500 s, every 10 s starting at 1000 s, and every 1000 s starting at 5000 s. Each record will contain the time, the pressure in volume CV200, the value of control function name CF4, and the total mass of hydrogen released in cavity name CAV3.

The difference between a WRITE file and a PUSH file is that in the latter the values of NCHAN data channels must be defined another *package*. MELGEN will terminate with an error message unless another package defines all of the data channels for a PUSH file during initialization. Similarly, MELCOR will terminate without completing the timestep on any step during which a new value for one or more channels fails to be defined before the end of the advancement. Thus, specific coding must be included in another package to transmit data directly to EDF. At this time, only the TP package has the capability to push data to EDF. It can record data about material parcels transferred to TP by an *in* process in a file for later use.

An example of input for such a PUSH file is:

```
EDF_ID   MELT-EJECTION   PUSH   MELTEJ.DAT
EDF_CHN   16
EDF_FMT   5E15.7   ! Each logical EDF record requires 4 lines
EDF_DTW   1 ! NT    TWEDF  DTWEDF
              1     3000.    10.
```

This defines an external data file with the user-defined name MELT-EJECTION, and in conjunction with the record

```
TP_INPUT
TP_IN  1 ! N NameTP   TYPE   MODE     NameEDF      NMSIN   NTHRM
           1  TP100   EDF    PUSH   MELT-EJECTION
```

will cause formatted records to be written to a file MELTEJ.DAT every 10 s, starting at 3000 s. (This file may be used as input to define debris sources in a later MELCOR run. A second, similar file would be required to record the associated radionuclide transfers and define sources in the second run.) See the description of input record TP_IN in the TP Users' Guide for more information.

7 Discussion of Output

The output from the EDF package will be self-explanatory. Each file will include a user-defined name, the transfer mode, and the values of all data channels. The channels for WRITE files are labeled by the control function argument involved, those for PUSH channels by names defined by the PUSH package, or by the label UNDEFINED*PUSHED*DATA if no other was defined. The channels for READ files are not labeled, as there is no way of knowing their contents.

Additional output provided for the first edit of each run gives the system name of the file, buffer information for READ files, and write increment control information for WRITE and PUSH files.

Appendix 1

MELCOR 2.0	PTFRead	Comments
EDF(NameEDF,m) NameEDF – the external data filename; m – the data channel number.	EDF.n.m n – the external data file index; m – the data channel number.	Value of the mth data channel in external data file NameEDF (index n).

Executive (EXEC) Package
Users' Guide

The MELCOR Executive (EXEC) package is responsible for overall execution control of MELGEN and MELCOR calculations. EXEC coordinates various processing tasks for other MELCOR packages, including file handling, input and output processing, sensitivity coefficient modification, system timestep selection, time advancement, and calculation termination.

This document describes general input philosophy and EXEC package input required for MELGEN and MELCOR. Control directives for input processing are covered, and generic instructions for modification of sensitivity coefficients, defined by other packages, are given. Specific input requirements for phenomenological models in other packages are described in the users' guides for those packages. An introduction to the capabilities of MELCOR, its overall philosophy and design, and the MELCOR documentation set is given in the MELCOR Primer.

Contents

CONTENTS **2**

1 INTRODUCTION **5**

1.1 Changes for MELCOR 2.1 7

1.1.1 Input File Structure 7

1.1.2 Records Types 8

2 MELGEN/MELCOR INPUT FORMAT **13**

3 INPUT PROCESSOR DIRECTIVES AND ENVIRONMENTAL DATA **15**

3.1 Input Processor Directives 15

ALLOWREPLACE – Allow Replacement Records 15
PROGRAM – Record to Allow Input to Many Codes in Same File 16
INCLUDE – Record to Redefine Input File 16

3.2 Named Comment Blocks 18

3.2.1 Introduction 18

3.2.2 Command Line 19

3.2.3 CommentBlock record 19

3.2.4 Interactive Method 19

3.2.5 Default Named Comment Blocks 20

3.3 Variable Field Names 20

3.3.1 Introduction 20

3.3.2 Variable Input File 21

3.3.3 VariableValue record 21

3.3.4 Interactive Method 21

3.4 Global Data 21

CommentBlock – List of comment blocks to explicitly activate 21
DefaultNamedCommentBlock – List of comment blocks to activate by default. 22
DefaultDirectory – Default directory for output files. 22
VariableValue – Defines a value for to be used for variable fields 23
DefineVariablesF*ile – Filename for defining values for variable fields 23
MEG_DIAGF*ILE – Filename for MELGEN Run Diagnostic Output 23
MEL_DIAGF*ILE – Filename for MELCOR run diagnostic output file 24
MEL_HTML*FILE – Filename for MELCOR run HTML output file 24
MEG_HTML*FILE – Filename for MELGEN run HTML output file 25

MEL_INSTALL – Directory location of MELCOR javascript files to be used for HTML output 25
MEG_OUTPUTF*ILE – Filename for MELGEN Output Listing File 25
MEL_OUTPUTF*ILE – Filename for MELCOR Output Listing File 26
RESTART*FILE – Filename for MELGEN/MELCOR Restart File 26
MEG_RESTARTF*ILE – Filename for MELGEN Restart File 28
MEL_RESTARTF*ILE – Filename for MELCOR Restart File and Restart Dump Number Used to Start Calculation 28
MEL_RFMOD – Restart File Modification Option 29
PLOTF*ILE – Filename for Plot File 30
PrintDefaultSC – Print Sensitivity Coefficient Default Values 31
PrintCurrentSC – Print Sensitivity Coefficient Current Values 31
MESSAGEF*ILE – Filename for Event Message Output 32
STATUSF*ILE - Filename for MELCOR Status Request File 32
STOPF*ILE – Filename for MELCOR Stop File 32
T_CON*TOUR – Filename for Temperature contour files 33
WRITENEWINP – Write New Input File in Canonical Form 33
EXTDIAGF*ILE – Filename for Extended Diagnostic Output File 34
RN1VISUALFILE – Filename for RN Visualization Output File 34
NOTEPAD++ – Enables NotePad++ edititing characters in text output. 35
PrintInputRecords – Generates a complete list of input directives known to MELGEN. 35
KEYWORDF*ILE – Generates a file containing plotvariable keywords formatted for ATLAS. 36
EXCELF*ILE – Currently unused. 36

4 MELGEN INPUT 36

EXEC_INPUT – Start Record for EXEC User-input 36
EXEC_DTTIME – Initial Timestep 36
EXEC_JOBID – Job identifier 37
EXEC_PLOT – Control Function Argument to be Plotted 37
EXEC_PLOTLENGTH – Fixed Length of Plot Key Name 38
EXEC_RUNONLY – Run Only On Specified Code Version **Error! Bookmark not defined.**
EXEC_TITLE – Title of the Calculation 39
EXEC_TSTART – Initial Start Time 39
EXEC_UNDEF – Redefine Initialization of Real Database and Scratch Storage 39
EXEC_GLOBAL_DFT – Global Default Scheme Record 40
EXEC_SS – Use Accelerated Steady State 41
EXEC_WRT - Write temperature file for COR and HS temperatures 41

5 MELCOR INPUT 42

EXEC_INPUT – Start Record for EXEC MELCOR User-input 42
EXEC_CFEXFILE – User specified CHANGEDATE File 42

EXEC_COMTC – Number of Matching Characters in MELGEN and MELCOR Titles 43
EXEC_CPULEFT – Desired Minimum Number of CPU Seconds Left at End of Calculation 44
EXEC_CPULIM – Maximum Number of CPU Seconds Allowed for this Execution 44
EXEC_CYMESF – Cycle Message Frequency Parameters 45
EXEC_DTINCR – Timestep Increase Factor 46
EXEC_DTMAXCF – Time Step Limit Control Function 46
EXEC_DTSUMMARY – Additional Timestep Data to Output File 46
EXEC_DTTIME – Initial Timestep 47
EXEC_EDITCF – Special Edit Control Function 47
EXEC_EXACTTIME – Define Exact End-of-time-step Times 48
EXEC_FORCEPLOT – Control of Extra Plot Information Following Timestep Cuts 48
EXEC_JOBID – Job Identifier 49
EXEC_PLOT – Control Function Argument to be Plotted 49
EXEC_NOCOPY – Suppress Copy of Files to Output File 50
EXEC_NOFLUSH – Suppress Explicit Buffer Flushing 50
EXEC_PLOTCF – Special Plot Control Function 51
EXEC_RESTARTCF – Special Restart Control Function 51
EXEC_RUNONLY – Run Only on Specified Code Version. **Error! Bookmark not defined.**
EXEC_SOFTDTMIN – Define Conditions for Limited Under-run of DTMIN Input 51
EXEC_STOPCF – Special Stop Control Function 52
EXEC_TIME – Timestep, Edit, Plot, and Restart Control 53
EXEC_TEND – End of Calculation Time 54
EXEC_TITLE – Title of the Calculation 54
EXEC_UNDEF – Redefine Initialization of Real Database and Scratch Storage 55
EXEC_WARNINGL*EVEL – Define Level for Warning and Information Messages 56

6 MELCOR SENSITIVITY COEFFICIENT MODIFICATION 57

7 PLOT VARIABLES 58

8 CONTROL FUNCTION ARGUMENTS 58

9 MELGEN AND MELCOR EXECUTION 59

10 MELCOR INTERACTIVE INTERRUPTS 61

11 EXAMPLE INPUT 62

1 Introduction

MELCOR executes control of calculations in two parts. The first part is a program called MELGEN, in which most of the input is specified, processed, and checked. When the input checks are satisfied, a restart file with all the information in the MELCOR database is written for the initial conditions of the calculation. The second part of MELCOR is the MELCOR program itself, which advances the problem through time by using the MELGEN database and any additional MELCOR input.

MELGEN and MELCOR share a structured, modular architecture that facilitates the incorporation of additional or alternative phenomenological models. This structure consists of four primary levels:

(1) The top level is the Executive (EXEC) package consisting of subroutines that coordinate the execution of various processing tasks and control the overall calculation sequence.

(2) The second level is composed of the database manager routines that pass the required portions of the MELCOR database to the third level.

(3) The third level contains the various modules or packages. Each package models a well-defined group of closely related phenomena or performs a well-defined special function. A unique two- or three-character alphanumeric identifier is assigned to each package and incorporated into the names of its subroutines and common blocks.

(4) The fourth level consists of routines performing utility-level functions such as matrix inversion, ordinary differential equation (ODE) integration, or transfer of mass and energy between packages. Many of these routines are contained in the MELCOR Utility (UTIL) package, which requires no user input.

Most of the computational effort in MELGEN takes place during three passes of input processing.

(1) The EXEC package calls each physics package to (1) read all the user-defined data for that package, (2) recalculate its database pointers for the size of the problem defined by the input, and (3) perform as much data initialization and error checking as is practical while using only data within that package.

(2) The EXEC package calls each physics package to (1) complete the initialization of its database with data defined in other packages during the first pass and (2) perform consistency checks with data from other packages.

(3) Initialize control functions that may depend on package data arguments defined during the second pass. See the Control Function (CF) Package Users' Guide.

Input processing proceeds completely through each pass, but the next pass is not attempted if errors have been detected during a pass. After all input has been processed to the extent allowed, EXEC calls each package to generate printed output and, if no fatal errors have been detected in any pass, to write the initial database into the restart file.

MELCOR execution involves two steps:

(1) Setup mode: In this step the database (generated at time zero by MELGEN or at a later time by a previous MELCOR run) is read from the restart file and any additional input is processed.

(2) Run mode: This step advances the simulation through time, updating the time-dependent portions of the database during each cycle and periodically writing the updated database into the restart file.

During setup mode, many of the tasks performed in MELGEN are repeated, including file opening, input preprocessing, sensitivity coefficient modification, and EXEC input processing. After the database is read from the restart file, any additional package input is read, processed, and checked with the database for consistency.

The heart of MELCOR is, of course, the problem advancement phase. The EXEC package first checks to see whether the calculation has reached an endpoint or the point where it generates the next output edit, plot dump, or restart dump. If the calculation has reached either point, each active package is called in turn to perform the appropriate task. EXEC then calculates the new system timestep (fixed for all packages), subject to the minimum and maximum specified by user input and according to requests by the packages themselves from the previous timestep.

Advancement across the EXEC package timestep proceeds in three stages, with a loop through all active packages in each stage:

(1) Allow all packages to initialize any interface of communication arrays needed by other packages during the advancement.

(2) Perform the time advancement for each package, any of which may elect to subcycle across the EXEC timestep using smaller steps. A package may also elect to call for a fallback, in which case EXEC repeats the entire current cycle with a smaller timestep.

(3) Allow each package to examine the results of calculations by other packages during the second stage. If calculated changes are excessive, a fallback may be called for at this point.

MELCOR's restart capability is extremely flexible. Each restart dump in the restart file contains all parts of the problem database required to restart MELCOR and continue the calculation. Restart dumps are generated by MELCOR at intervals specified by the user based on either simulation time or central processing unit (CPU) time and are generated

at the end of the calculation. Special restart dumps are also generated at key events such as vessel or containment failure. Sensitivity coefficients and selected MELCOR input variables can be changed at a restart to examine different calculation scenarios from that point, thus eliminating the need to repeat portions of the calculation that do not change. MELCOR restarts are exact; that is, a calculation restarted at some point with no change in input will continue along the identical calculation path that the code would have taken if it had not been stopped and restarted at that point.

One of the most important output files generated during a MELCOR run is the plot file, which contains data necessary to plot time histories for many of the variables calculated during the advancement. Although the volume of data involved is already substantial, there are many variables that a user might wish to plot but which are not included in the file because their inclusion could make the file unmanageably large. Many of these variables are available as control function arguments that are not directly plottable. In the past, users have been forced to deal with such variables by defining control functions "EQUAL" to the corresponding arguments. The values of these control functions, which are automatically included in the plot file, can then be plotted. See the CF Package Users' Guide for details.

Although effective, this approach is cumbersome and often requires the definition of hundreds of control functions. Furthermore, because all control functions must be defined in MELGEN, there is no way to add a variable to the plot file without rerunning the entire calculation. Therefore, a capability has been developed to allow the user to (1) directly define a list of control function arguments that can be added to the plot file in MELGEN and (2) add to that list at the start or at any restart of MELCOR under EXEC package control. The necessary input is defined in Sections 4 and 5 of this users' guide, under the description of the EXEC_PLOT record.

1.1 Changes for MELCOR 2.1

A number of extensions and enhancements to the models have been made to the EXEC package in MELCOR 2.1. They are summarized in the following subsection.

1.1.1 Input File Structure

Input deck structure is aligned with the object-oriented approach for data storage and processing.

All input deck records are divided according to the structure of the object-oriented database:

program level → package level → object level

- Program level contains data on all packages.
- Package level contains general data for the whole package.
- Object level contains data records for sub-objects in the package.

- Identification of the objects:

  on program level:

  PROGRAM <program name>
  END PROGRAM < program name>

  on package level:

  XYZ_INPUT <additional option>

  on object level: user-input number → unique user-defined name

  XYZ_ID <object name> <data fields>

The identification record must be placed before all data records for a given object. The data records within one level of responsibility may have an arbitrary order.

The following example shows one possible input file structure:

```
! some comments
! input processor directives and environmental data
...
!
PROGRAM MELGEN                                ! program id record
...
   ! data block for EXEC package ***
   EXEC_INPUT                                 ! package id record
   EXEC_JOBID 'TEST2'                         ! package data
   EXEC_TITLE 'TEST2'                         !
   ...
   ! data block for FL package *****
   FL_INPUT
   !*  1 -  FL data block **********
     FL_ID  FL31                              ! object id record
     FL_FT  CV120 CV300 0.00  -0.20           ! object data
     FL_GEO 0.10E+01  0.20  0.0   0.0         !
     ...
   !*  2 -  FL data block **********
     FL_ID  FL115                              ! object id record
     FL_GEO 0.70E+01 0.40E+01 0.10E+01 0.00    ! object data
     FL_FT CV110 CV150    0.10E+02   0.10E+02 !
     ...
   ...
...
END PROGRAM MELGEN
PROGRAM MELCOR
...
END PROGRAM MELCOR
```

In this input fragment, data blocks for each level of responsibility are indented only to illustrate the file structure. The user is not required to indent the data blocks.

1.1.2 Records Types

All record specifications have one of the following general forms:

- Start and end record for the program:

  PROGRAM <program name>
  END PROGRAM < program name>

- Start record for each package:

  XYZ_INPUT <package activity>

- Data record:

  Record with the object name:

  XYZ_ID <object name> <field1> ... <fieldN>

  Simple data record:

  XYZ_ABC <field1> ... <fieldN>

- Tabular record

- Record with the tree structure.

Package start record has the following structure: XYZ_INPUT, where XYZ is the corresponding package mnemonic. The name of the start record for each package is predefined:

BUR_INPUT	CAV_INPUT	CF_INPUT	COR_INPUT
CVH_INPUT	DCH_INPUT	EDF_INPUT	ESF_INPUT
EXEC_INPUT	FDI_INPUT	FL_INPUT	HS_INPUT
H2O_INPUT	MP_INPUT	NCG_INPUT	RN1_INPUT
RN2_INPUT	SPR_INPUT	TF_INPUT	TP_INPUT

The general format for data records is as follows:

XYZ_ABCnn <value of field1> <value of field2> … <value of fieldN>

The data record name has the following general form:

XYZ_ABCnn,

where

XYZ = a mnemonic for the package that this data record belongs to;

ABC = a character mnemonic for a full record name; the characters suggest the record's content or the object that these data belong to;

nn = additional index added to some record names; for example, when the data for some models or objects are arranged for a series of records. These indices have the alphanumeric format.

Although the naming structure of all records is similar, the name of each one is predefined and cannot be changed by the user.

The data record can contain fields for integers, reals, or character data. Reals should contain <.> (point sign) and may also contain a sign and/or order part <eE>. The field can be required or optional. The user must input a required field according to the field specification. Optional fields are usually placed at the end of the record and may be omitted; default values will then be used.

The format of each MELGEN or MELCOR input record is specified in the users' guide for the package to which that input record belongs. For example, the specifications for records describing hydrodynamic control volumes will be found in the Control Volume Hydrodynamics (CVH) Users' Guide while specifications for records describing control functions will be found in the CF Users' Guide.

All record specifications have the same general form. For example, a hypothetical input record might be specified as

XYZ_ABC - Data Definition
Optional

(1) NAME
Name of some object.
(type = character*16, default = none)

(2) VAR
Value of a variable names "VAR."
(type = real, default=0.0, units = kg/s)

This describes a record XYZ_ABC in which the user may define one or two fields. The first is a CHARACTER name with a maximum length of 16 characters; the second, which is optional, must contain a real number if the value is specified. If absent, a value of 0.0 will be assumed.

Certain types of data records have unique input features and require additional description. These record types include the object definition record, the simple data record, the tabular record, and the record with a tree structure.

The *object definition record* usually has the XYZ_ID name, where XYZ refers to the package mnemonic that this object belongs to. The format for an object definition record is

XYZ_ID <the object name> <additional field(s)>

The user-defined number is no longer required; therefore, the object name must be defined without fail. Object names for each particular package must be unique because these names are used for connections. However, for different packages the names may be repeated. The name of the object is a required parameter, and this record must be the first input in the data block for the specified object. The name input on the first field of XYZ_ID record identifies the object. The data are input on the additional fields in this record or in records that follow.

The *simple data record* contains one or more data fields—some required and others optional. If the record has at least one required field, the user must input the required data. If all fields are optional, then the record is optional too and may be omitted. In addition, a record may be optional if the data it contains are needed only in particular cases (for example, for a special model). The format for such records is

```
XYZ_ABC <req.value1> … <req.valueN> <opt.value1> … ! comment
```

The *tabular records* allow the user (1) to input similar or seriated data as a table where data lines can be replaced or redefined; and (2) to define a set of identical objects. Each tabular record consists of two parts: the head of the record and the data lines. General record format is as follows:

```
! the head of record
XYZ_ABC <dimension> <additional field(s)>   ! comment
! data lines
<integer identifier of string> <field1> … <fieldN>   ! comment
<integer identifier of string> <field1> … <fieldN>   ! comment
```

The head of each record contains the record name and the table dimension. The head may also have some required and/or optional fields along with comment fields. The dimensions must exactly equal the number of data lines in the table. An integer identifier must be input at the beginning of each data line. The integer identifier may serve as a simple sequencing parameter or as an informative identifier when the ALLOWREPLACE directive is active.

The following example shows specifications for the above record:

XYZ_ABC - <Record's name>
Required (if) / Optional

<Notes on the record content>

(1) DIM
Length of the table.
(type = integer, default = , units =)

(2) ADDFLD
<Additional field name>
(type = , default= , units =)

The following data are input as a table with length DIM:

(1) INTID
Integer table line identifier.
(type = integer, default = , units =)

(2) FLD1
<Field name>
(type = , default= , units =)

…

(N+1) FLDN
<Field name>
(type = , default= , units =)

When tabular records define a set of identical objects, each line of the table contains the data for a separate object. The name of this object is input into a required field. For example, valve input in the FL package can be specified as the following.

```
FL_VLV 5 ! NV  VLVNAME  FLNAME  KEYTRIP  NVFONF  *NVFONR  NVTRIP
           1   VALVE1   FL31    NoTRIP   CF130
           2   VALVE2   FL199   NoTRIP   CF199
           3   VALVE3   FL230   NoTRIP   CF232
           4   VALVE4   FL398   UseTRIP  CF502    CF502   CF501
           5   VALVE5   FL399   UseTRIP  CF402    CF402   CF401
```

Field VLVNAME is a required field because it is used to define an object. Other fields in the corresponding line define the data for the object. These fields may be either required or optional. An integer identifier may be used to replace data (if the ALLOWREPLACE directive is active).

Record with a tree structure is used to divide a set of input records or fields within the record itself according to the input logic. Additional keys and switches are used to specify the branches of tree. Keys and switches are input as separate variables, not as parts of other variables. Character equivalents are added to most parts of switches and keys. The following simple example (from Heat Structure (HS) package) shows typical input of control and tabular functions:

HS_SRC – Internal Power Source Data
Required

(1) ISRC
Internal power source type.
(a) -1 or CF
Power is given by control function.
(b) 0 or NO
No internal power source exists for this heat structure.
(c) 1 or TF

Power is entered as a function of time in tabular function.
(type = integer / character*2, default = none)

(2) NAMESRS
Name of CF (if ISRC equals –1 or 'CF')
Name of TF (if ISRC equals 1 or 'TF').
This field is required if ISRC equals –1/'CF' or 1/'TF'.
(type = character*16 , default= none , units = none)

:

Examples

```
HS_SRC NO           ! no power source
HS_SRC -1   CF500   ! use cf
HS_SRC TF   TF201   ! use tf
```

2 MELGEN/MELCOR Input Format

Commands to MELGEN and MELCOR are contained in one or more input files. The filename is entered at the code prompt or (on most computer systems) as a parameter on the execution line.

A free-format is used to preprocess all input. Every record must either have a valid identifier field or must be an input directive record, which controls overall processing options; a comment record; or the terminator record.

The leftmost field is interpreted as an identifier field. Duplicate or replacement records are discussed below.

Only alphanumeric characters may be used for identifier fields (with the exception of integer identifiers in tabular records). User input data fields may be integers, reals, or character data. For example,

IDENT 0 1. 1.E28 1.D29 'CHARACTER DATA'

Delimiters are blanks (·) or tab (→) characters. Multiple delimiters have the same effect as one delimiter. For example, the records noted below are identical:

IDENT·····4.4·····5.5·····6.6
IDENT → 4.4 → 5.5 → 6.6

Character data may be enclosed by single quotes. If there are no internal blanks or commas and if the string is not interpreted as a number, then the single quotes may be left off. For example, the following record contains an identifier field and four character data fields:

IDENT BOY 'BIG BOY' '1.1' 'MY,FAVORITE,YEAR'

All input is case insensitive. Lower case characters are automatically converted to upper case, except for character strings enclosed by single quotes. For example, the following three statements are identical:

ident george 1.e7
IDENT GEORGE 1.E7
Ident George 10000000.

When single quotes are used, however, the situation changes. In the following examples the parameters are all different:

ident george filename
IDENT 'george' 'filename'
Ident 'George' 'Filename'

In some situations MELCOR uses multiword parameters. In general the input processor will accept these quantities with a blank separator in single quotes or separator formats. For example, "STAINLESS STEEL OXIDE" can be referenced in a heat structure as

'STAINLESS STEEL OXIDE'
stainless-steel-oxide
Stainless-Steel-Oxide

But it is safer to use identifiers without spaces and single quotes.

In general, lower case characters should not be used in multiword parameters inside single quotes because they will then not be converted to upper case and, therefore, may not match the appropriate identifier or filename. See individual package users' guides for details.

A comment record has an exclamation mark (!) as the leftmost nonblank character. All characters to the right of an exclamation are interpreted as comments. Completely blank lines are considered comments. An example of a comment line is

! THIS RECORD IS JUST A LITTLE OLD COMMENT

A comment field may follow a noncomment field on the same record. For example,

IDENT 4.4 5.5 ! FORWARD AND REVERSE LOSS COEFFICIENTS

An entire part of input deck enclosed in triple parenthesis placed as the leftmost nonblank characters— (((and))) —is regarded as a comment and ignored when processing the input deck. For example,

```
NCG_INPUT !********************
! these records are processed:
NCG_ID N2        !    4
NCG_ID O2        !    5
NCG_ID H2        !    6
! and these ones are comments:
```

```
(((
NCG_ID CO2       !    7
NCG_ID CO        !    8
NCG_ID CH4       !    9
)))
! END NCG input***************
```

In some cases the identifier records allow optional trailing characters. This situation is denoted in this manual as

MEG_RESTARTF*ILE

For this input form, the code will recognize MEG_RESTARTF and MEG_RESTARTFILE as the same input.

The default processing of duplicate records (those with the same identifier field) generates a fatal error. An input processor record ALLOWREPLACE permits a duplicate record to replace any and all records with the same identifier field that appears earlier in the input stream.

3 Input Processor Directives and Environmental Data

There are several input commands, termed input processor directives, which control the processing of other input. For the most part, these commands appear as any other command except when an error occurs.

3.1 Input Processor Directives

ALLOWREPLACE – Allow Replacement Records

Optional

Records with the same identification field are duplicates. The default is to treat these duplicates or replacement records as a fatal error. This command allows replacement records where only the last record in the input set with the duplicate field is retained. A warning message is generated by replacement.

The intended use of the replacement record feature is to permit changes to QA controlled or otherwise protected input sets without having to edit the input itself. Sensitivity studies and other changes can be completed without disturbing the base case input set. When this feature is desired, the ALLOWREPLACE command is required.

This record must be placed in the input set before any replacement records; it is strongly recommended to place it at the beginning of the input set.

PROGRAM – Record to Allow Input to Many Codes in Same File

Optional

Input sets are often moved between different computers and users in modern computer environments. All related input files for MELGEN, MELCOR, and other related programs can be combined into a single computer file by using the program name directive. The last record in an input stream for a program must be the terminator record END PROGRAM NAME. This command is designed to ensure that all the different input sets for a single problem do not get separated.

The file structure is

```
PROGRAM MELGEN
        :
!   MELGEN input
        :
END PROGRAM MELGEN
PROGRAM MELCOR
        :
!   MELCOR input
        :
END PROGRAM MELCOR
PROGRAM anything else
        :
    Ignored by MELGEN, MELCOR
        :
END PROGRAM
{end of file}
```

The record is case insensitive. For example, MELCOR will process only data following a PROGRAM MELCOR up to the END PROGRAM record or end of file. Multiple PROGRAM records with the same name are allowed. The only exclusion is allowed outside PROGRAM / END PROGRAM scope—ENVIRONMENT input data, which should start input deck.

If PROGRAM records are not used in input, all records in the input file are ignored and an error message will be produced.

INCLUDE – Record to Redefine Input File

Optional

(1) RIFILE

Filename for additional redirected input file.

(type = character*80, default = none)

The user may define a data block in the above-mentioned input file to be read in. In this case, the data block must be specified in the input file by the character string BLOCK: followed by the user-defined name of this block (exactly as BLNAME).

(2) BLNAME

User-defined name of the data block in the RIFILE to be read in.

(type = character*80, default = none)

The include command allows the input set to be spread over multiple files. This record has the identifier field INCLUDE followed by a filename. The file is read into the input stream at the current position. After the new file's data are finished, the new file is closed and reading continues with the next record in the original file. INCLUDE directives may be nested at any layer.

Optional parameter <block> is used for insertion of the separate block from file. The parameter <block> can be set as the third optional field after include directive and filename.

INCLUDE <file> <block>

The line

BLOCK: <block>

should exist in the included file. Information from the included file from one BLOCK: to the next will be inserted to the head file.

An input file might look like the following:

Example

```
:
INCLUDE    CVHDAT  ! CVH INPUT DATA
INCLUDE    CORDAT  ! CORE INPUT DATA
INCLUDE    RNDAT   ! RADIONUCLIDE INPUT DATA
INCLUDE    HSDAT   ! HEAT SLAB INPUT DATA
:
INCLUDE    'HSFT.INP'    HS40019        ! USE DATA BLOCK FROM FILE
```

This example shows include statements in the MELGEN input file:

```
! Head file
CVH_INPUT
  CV_ID  'Pipe3'
  CV_VEL 0.23  0.56
  ...
INCLUDE 'CVH_HS.INP' 'UPPER PLENUM'
….
! CVH_HS.INP FILE
. . .
BLOCK: 'UPPER PLENUM'
CV_ID 'UPPER PLENUM ' 'CVT2'
```

```
CV_THR Equil FOG ACTIVE
...
BLOCK: 'UPPER VESSEL'
HS_ID 'UPPER VESSEL'
HS_GD CYLINDRICAL      YES
...
```

3.2 Named Comment Blocks

3.2.1 Introduction

This feature was added to allow the user to include blocks of input records which are selected from a variety of input methods. It can be used for quality assurance purposes to test variations of input records or it can be used to encapsulate multiple accident variations or sensitivity calculations within a single input deck. Alternate input options can be easily activated or deactivated on input, where the selection can be made from a command line argument, making it possible to automate testing of multiple options.

MELCOR comment blocks were added as part of the code conversion project. In MELCOR 2.1, a block of input can be commented out by enclosing that block of input within a set of triple parenthesis, i.e.,

```
(((
…  !These input records are ignored
)))
```

This structure has been extended to allow named comment blocks that can either be ignored or read by MELGEN/MELCOR depending on the appropriate input. For example, the following demonstrates the necessary input for MELGEN to read a block of records required for an SBO calculation. For this case, only the SBO comment block is read and the LBLOCA block is ignored. Alternatively, if the second field on the comment block record were LBLOCA, the LBLOCA records would be read. The CommentBlock record can also contain more than one field so that multiple comment blocks can be read.

```
CommentBlock  SBO
...
PROGRAM MELGEN
...
(((SBO  ! Additional comments placed here
...    !These input records are not ignored
)))
...
(((LBLOCA   ! This is a large break LOCA scenario
...    ! These input records are ignored
)))
```

There can be multiple comment blocks with the same name which can be useful if you have many test options with variations in multiple locations in the input deck (i.e., different volume altitude tables and different flow path options). Furthermore, the named comment block construct allows multiple names or aliases for a comment block. This would allow some commonalities between a subset of multiple test options.

```
CommentBlock  SBO
```

```
...
PROGRAM MELGEN
...
(((SBO  !Additional comments placed here
...These input records are not ignored
)))
...
(((LBLOCA   !This is a large break LOCA scenario
...These input records are ignored
)))
(((SBO
...These input records are not ignored
)))
(((SBO+LBLOCA
...These input records are not ignored for either SBO or LBLOCA cases
)))
…
```

In addition to this method of selecting named comment blocks to be uncommented, named comment blocks can be selected interactively or from command line input. The precedence for the selection method is as follows:

3.2.2 Command Line

Comment blocks to be uncommented are provided in a command line argument. If the command line argument is present, this method takes precedence over other methods.

```
melgen i=inputfile c=case1+case2+case3
```

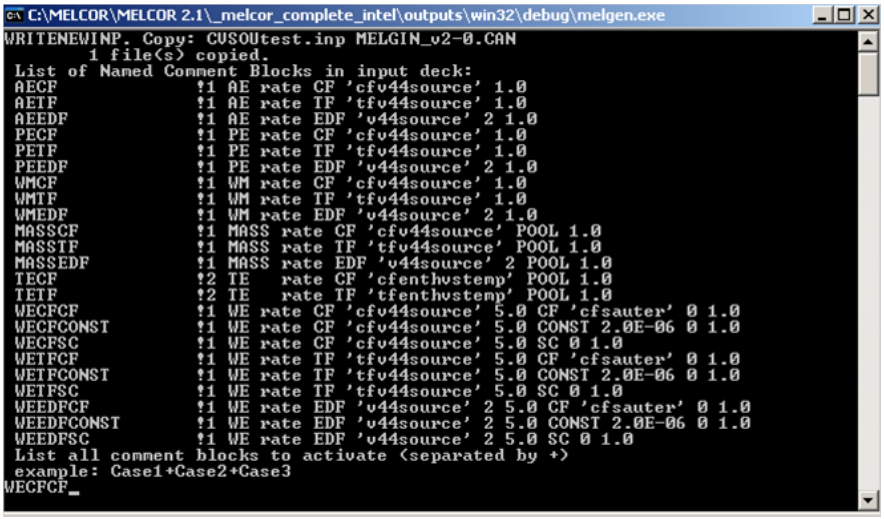
3.2.3 CommentBlock record

Comment blocks are selected from the CommentBlock record discussed above.

```
! These 3 named comment blocks are uncommented
CommentBlock case1 case2 case3
```

3.2.4 Interactive Method

A list of comment blocks available in the input deck is printed to the screen and the user types in a list of all comment blocks desired. Note that this method is only available if the commandline argument and the CommentBlock record are absent. A screenshot showing the interactive interface is shown below:



In this example, 23 test cases are encapsulated within a single input file and individual cases (or combinations of cases) can be selected from input. Note also that comments in the input deck are echoed in the interactive list of named comment blocks providing additional information to the user. Multiple named comment blocks can be activated by separating the block names with a '+' sign.

3.2.5 Default Named Comment Blocks

The user can also specify a number of named comment blocks to un-comment by default. The DefaultNamedCommentBlock record is placed in the global variable section before all program input. If the user is prompted for a list of named comment blocks to activate and the default record is present, and the user returns without entering in any values, then the named comment blocks on this record are used by default. If the user otherwise specifies any named comment blocks to uncomment, this record is ignored.

```
DefaultNamedCommentBlock 2HSNODE+BIGDT
```

3.3 Variable Field Names

3.3.1 Introduction

This feature allows the user to insert variable input fields into records which can then be updated through user input at the time of execution. This can be useful for testing ranges of input fields, managing variant assessment calculations, steady state initialization and could be used to aid interfacing MELCOR to other applications. Users can provide variable values through a variety of methods.

Input fields are tagged so that MELCOR will preprocess the input file, replacing the tagged variables with a value specified by the user. These tagged variables can occur multiple places in the input deck and all instances are replaced by the same value. The tagged placeholder on an input record includes the variable name and the default value enclosed in a set of triple curly parenthesis, {{{}}}, i.e.,

```
FL_SEG 1        !        SAREA            SLEN            SHYD        SRGH    LAMFLG   SLAM/CFNAME
                1  {{{SAREA=1.0}}}  {{{SLEN=0.2}}}   0.24000E-01  0.5E-04   CONST      0.00
```

In this example, two variables, SAREA and SLEN, are specified on the flow segment record. Default values are 1.0 and 0.2 respectively are provided. In addition to this method of providing variable values, values can be selected interactively or from command line input. The order for the various methods of specifying values for variables is as follows:

3.3.2 Variable Input File

A file can be specified containing values for all or some of the variables found in the input deck. If some variable values are missing from this file, the user will be asked interactively to provide values. A record in the global specification section of the MELCOR input deck can be used to provide the name of this file. If this file is missing, the user will be asked to interactively provide values.

```
DefineVariablesFile    Variables.dat  !Values read from Variables.dat file
```

If this record is present and the specified file is missing, an error message will be issued and the problem will terminate. The format for the Variables.dat file is as follows:

```
{{{Var1=1.0}}}        !Additional comment
{{{Var2='Wetwell'}}}  !Additional comment
{{{Var3=3}}}          !Additional comment
```

3.3.3 VariableValue record

Variable values are provided on the VariableValue record to be located in the global variable section of the input prior to program input. Note that any data type can be provided but must be consistent with the requirements of the associated input record to avoid MELGEN errors. The supplied value effectively replaces the {{{var=nnn}}} variable field.

```
!These 3 variables are evaluated
VariableValue {{{Var1=1.00}}} {{{Var2='Wetwell'}}} {{{Var3=3}}}
```

3.3.4 Interactive Method

A list of all variables is provided and the user is queried for the required value. If the user hits return without specifying a value, then the default value will be used for this value unless it was specified by any of the previous methods.

3.4 Global Data

The Global data block should contain the names of files you need to read or write.

All global records should be placed at the start of input deck in any order.

CommentBlock – List of comment blocks to explicitly activate

Optional

The user can also specify a number of named comment blocks to un-comment by default. The CommentBlock record is placed in the global variable section before all program input. The user can specify a list of all named comment blocks that will be

activated. If a block is listed on this record, then this value will be used and the user will not be queried interactively for a value and the default value will not be used.

(n) BlockNames

List of block file names to activate by default

(type = character*80, default = none)

Example

```
CommentBlock AFW ACCUM HHSI
```

DefaultNamedCommentBlock – List of comment blocks to activate by default.

Optional

The user can also specify a number of named comment blocks to un-comment by default. The DefaultNamedCommentBlock record is placed in the global variable section before all program input. There will still be an interactive prompt for a list of comment blocks, but if the user returns without entering any values, then the named comment blocks on this record are used by default. If the user otherwise specifies any named comment blocks to uncomment, this record is ignored.

(n) BlockNames

List of block file names to activate by default (file names are delimited by '+' sign.

(type = character*80, default = none)

Example

```
DefaultNamedCommentBlock AFW+ACCUM+HHSI
```

DefaultDirectory – Default directory for output files.

Optional

By default, MELCOR places all output files in the same directory as the master input file. The user can specify an alternate directory location with this input record. If the directory does not exist, MELGEN will create one. The directory can be specified as a relative path, relative to the folder containing the input deck.

This record works only in a Windows environment.

(1) DefDIR

Provides the name of the default directory.

(type = character*80, default = none)

Example

```
DefaultDirectory '.\ HS21node\
DEFAULTDIRECTORY  'E:\CustomBuild\MELCOR_2x\test2\'
```

VariableValue – Defines a value to be used for variable fields

Optional

Fields on MELGEN or MELCOR input records can be defined as named variables and their numeric values can be specified on this record. Alternatively, these values are specified in the variable definition file (see DefineVariablesF*ile below) or specified interactively.

Example

```
VariableValue  {{{Var1=1.0}}} {{{Var2='Wetwell'}}}
```

DefineVariablesF*ile – Filename for defining values for variable fields

Optional

A file can be specified containing values for all or some of the variable fields found in the input deck. If some variable values are missing from this file, the user will be asked interactively to provide values. If this file is missing, and values are not specified on the VariableValue record, the user will be asked to interactively provide values.

If this record is present and the specified file is missing, an error message will be issued and the problem will terminate.

Example

```
DefineVariablesFile    Variables.dat  !Values read from Variables.dat file
```

The format for the Variables.dat file is as follows:

```
{{{Var1=1.0}}}        !Additional comment
{{{Var2='Wetwell'}}}  !Additional comment
{{{Var3=3}}}          !Additional comment
```

MEG_DIAGF*ILE – Filename for MELGEN Run Diagnostic Output

Optional

The diagnostic output from MELGEN is written to the file named on this record. If this record is not present, the diagnostic output is written to MEGDIA_v2-0.

(1) DIAGFILE

Filename for diagnostic output file.

(type = character*80, default = MEGDIA_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MEG_DIAGFILE BLOWDOWN_G_DIA.TXT
```

MEL_DIAGF*ILE – Filename for MELCOR run diagnostic output file

Optional

The diagnostic output from MELCOR is written to the file named on this record. If this record is not present, the diagnostic output is written onto MELDIA_v2-0.

(1) DIAGFILE

Filename for diagnostic output.

(type = character*80, default = MELDIA_v2-0)

Example

```
MEL_DIAGFILE BLOWDOWN_DIA.TXT
```

MEL_HTML*FILE – Filename for MELCOR run HTML output file

Optional

The text output file is converted to HTML output if this record is specified. An HTML file with this name will be found in the working directory and individual HTML files for each time edit will be found in a subfolder named 'HTML.' If this record is not present, no HTML files will be generated for the calculation.

(1) HTMLFILE

Filename for HTML output.

(type = character*80, default = none)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes. For access to all javascript menus and links, certain javascript files must be available (see MEL_INSTALL record in global variable description).

Example

```
MEL_HTML BLOWDOWN_OUT.HTM
```

MEG_HTML*FILE – Filename for MELGEN run HTML output file

Optional

Same as MEL_HTML*FILE but for MELGEN

(1) HTMLFILE

Filename for HTML output.

(type = character*80, default = none)

MEL_INSTALL – Directory location of MELCOR javascript files to be used for HTML output

Optional

Graphics capabilities and other enhancements available for HTML output files require certain javascript files that must be available. This record identifies the name of the directory that contains all necessary files and subfolders. These files will be loaded as part of the installation of the MELCOR LaunchPad.

(1) MELINSTALL

Path to the MELCOR javascript installation.

(type = character*80, default = none)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MEL_INSTALL 'C:\Program Files\MELCOR'
```

MEG_OUTPUTF*ILE – Filename for MELGEN Output Listing File

Optional

The listing output from MELGEN is written to the file named on this record. If this record is not present, the output is written to MEGOUT_v2-0.

(1) OUTPUTFILE

Alternate filename for all listable output.

(type = character*80, default = MEGOUT_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MEG_OUTPUTFILE BLOWDOWN_G_OUT.TXT
```

MEL_OUTPUTF*ILE – Filename for MELCOR Output Listing File

Optional

The listing output from MELCOR is written to the file named on this record. If this record is not present, the output is written onto file MELOUT_v2-0.

(1) OUTPUTFILE

Alternate filename for all listable output.

(type = character*80, default = MELOUT_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MEL_OUTPUTFILE BLOWDOWN_OUT.TXT
```

RESTART*FILE – Filename for MELGEN/MELCOR Restart File

Optional

The binary restart file for cycle zero is written to the file named on this record. If this record is not present, then the restart file is written to MELRST_v2-0.

(1) RESTARTFILE

Filename for restart file.

(type = character*80, default = MELRST_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes. The file that is specified will be used as the restart file regardless of whether MELGEN or MELCOR is the active program.

Every dump on the restart file is associated with a cycle number and problem time. All information required to continue the calculation is contained in each dump. The dump generated by MELGEN is always cycle zero. The frequencies of dumps generated by MELCOR are specified on the EXEC_TIME record. A restart file dump is always generated when the calculation terminates gracefully, either at the time

specified on the EXEC_TEND record or when terminated by a MELCOR-detected error from which it cannot recover. On some systems, a restart dump will also be successfully generated on abnormal terminations (i.e., system-generated aborts or bombs) through the use of system-dependent error recovery routines. The user may specify which dump in the restart file to use when starting the calculation. This allows the user to run part of the calculation, stop, and/or restart and continue the calculation. The user may also stop the calculation, change limited input parameters in order to run a variation in a sensitivity study, and continue the new calculation.

A second field is optional for MELGEN input, but required if used by MELCOR. There are two forms of the following input:

FORM 1: Cycle number input

(2) NCYCLE

The character string NCYCLE

(type = character, default = -1, units = none)

(3) NREST

(a) ≥ 0

Cycle number to start the calculation.

(b) -1

The last restart dump is used.
At each restart dump found, the user will then be queried on whether that is the desired restart point.

(type = integer, default = -1, units = none)

FORM 2: Calculation time input

(2) TIME

the character string TIME.

(type = character, default = none)

(3) PROBLEM_TIME

Calculation time to restart execution.
The code will restart from the first time on the restart file with time greater than or equal to the input.

(type = real, default = none, units = s)

Examples

```
RESTARTFILE BLOWDOWN.RST TIME 3600.0 ! restart at first dump * ≥ 3600. s
RESTART Case1                        ! Only valid for MELGEN
RESTARTF MyRestart NCYCLE  0         ! Valid for either MELGEN or MELCOR
```

MEG_RESTARTF*ILE – Filename for MELGEN Restart File

Optional

The binary restart file for cycle zero is written to the file named on this record. If this record is not present, then the restart file is written to MELRST_v2-0.

(1) RESTARTFILE

Filename for cycle zero restart file.

(type = character*80, default = MELRST_v2-0)

Though the restart file that is used by both MELGEN and MELCOR can be specified by the same RESTARTFILE record, it is possible to alternatively specify separate restart files for either MELGEN or MELCOR. Note that for this case the user must be very careful that the appropriate file is intended. Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MEG_RESTARTFILE BLOWDOWN.RST
```

MEL_RESTARTF*ILE – Filename for MELCOR Restart File and Restart Dump Number Used to Start Calculation

Optional

The binary restart file to restart the calculation is specified by the file named on this record. If this record is not present, then the restart file used is MELRST_v2 0.

(1) RESTARTFILE

Filename for restart file.

(type = character*80, default = MELRST_v2-0)

Though the restart file that is used by both MELGEN and MELCOR can be specified by the same RESTARTFILE record, it is possible to alternatively specify separate restart files for either MELGEN or MELCOR. Note that for this case the user must be very careful that the appropriate file is intended.

Every dump on the restart file is associated with a cycle number and problem time. All information required to continue the calculation is contained in each dump. The dump generated by MELGEN is always cycle zero. The frequencies of dumps generated by MELCOR are specified on the EXEC_TIME record. A restart file dump is always generated when the calculation terminates gracefully, either at the time specified on the EXEC_TEND record or when terminated by a MELCOR-detected error from which it cannot recover. On some systems, a restart dump will also be successfully generated on abnormal terminations (i.e., system-generated aborts or

bombs) through the use of system-dependent error recovery routines. The user may specify which dump in the restart file to use when starting the calculation. This allows the user to run part of the calculation, stop, and/or restart and continue the calculation. The user may also stop the calculation, change limited input parameters in order to run a variation in a sensitivity study, and continue the new calculation.

There are two forms of the following input:

FORM 1: Cycle number input

(2) NCYCLE

The character string NCYCLE

(type = character, default = n, units = none)

(3) NREST

(c) ≥ 0

Cycle number to start the calculation.

(d) -1

The last restart dump is used.
At each restart dump found, the user will then be queried on whether that is the desired restart point.

(type = integer, default = none, units = none)

FORM 2: Calculation time input

(2) TIME

the character string TIME.

(type = character, default = none)

(3) PROBLEM_TIME

Calculation time to restart execution.
The code will restart from the first time on the restart file with time greater than or equal to the input.

(type = real, default = none, units = s)

Examples

```
MEL_RESTARTFILE BLOWDOWN.RST TIME 3600.0 ! restart at first dump *          ⩾ 3600. s
MEL_RESTARTFILE MELRST NCYCLE -1     * use last restart dump
MEL_RESTARTFILE MELRST NCYCLE  0     * restart dump number
```

MEL_RFMOD – Restart File Modification Option

Optional

This record allows the user to copy selected restart dumps into a new restart file. User input history stored on the old restart file is also copied to the new file. This option allows the user to delete unnecessary restart dumps (i.e., dumps written at predetermined intervals for insurance but not actually needed to provide restart points) that merely consume file space. An arbitrary number of CYCLEN fields may follow the first field. No other input records are required if the EXEC_RFMOD record is present, but the MEL_RESTARTFILE record may be used to specify the old restart file.

(1) NRESTF

Filename for new restart file.

(type = character*80, default = none)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose in single quotes.

(N) CYCLEN

List of cycle numbers of restart dumps to be written to new restart file.
Values may be input in the form of a list of cycle numbers or as a range of numbers a–b (without any blanks). An arbitrary number of these may be included on this record. The field may be -1 to designate the final restart dump without specifying its cycle number. If you use the spaces you should enclose list in single quotes. You may use commas without spaces and single quotes.

(type = integer, default = none, units = none)

Examples

```
MEL_RFMOD 'filename' '1 3 5-15 –1'
MEL_RFMOD 'filename' 1,3,5-15,–1
```

PLOTF*ILE – Filename for Plot File

Optional

The binary plot data are written to the file named on this record. If this record is not present, the plot data are written onto the file MELPTF.PTF.

(1) PLOTFILE

Alternate filename for plot dump file.

(type = character*80, default = MELPTF.PTF)

(2) PLOTFORMAT

Switch for the plot file format.

(a) 0 or OLDFORMAT
The plot file will be written in the format recognized by PTFRead;

(b) 1 or NEWFORMAT
The plot file will be written in the format recognized by GUI.

(type = character*9, default = OLDFORMAT)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
PLOTFILE BLOWDOWN.PTF OLDFORMAT
```

PrintDefaultSC – Print Sensitivity Coefficient Default Values

Optional

Default values for all sensitivity coefficients in active packages are printed to the file named on this record. The default values printed are those defaults 'after' all global or package default records are processed, i.e., EXEC_GLOBAL_DFT, COR_DFT, etc. This report is printed during MELGEN and does not include user specification from XXX_SC input records. When this record is absent, no report is printed.

PrintCurrentSC – Print Sensitivity Coefficient Current Values

Optional

Current values for all sensitivity coefficients in active packages are printed to the file named on this record. This report is printed just after the MELCOR input is read and includes all user specifications from XXX_SC input records and any default set specifications (EXEC_GLOBAL_DFT, COR_DFT, etc.). When this record is absent, no report is printed. If a filename is not provided by the user, then the sensitivity coefficients are written to the file 'CurrentSC.out.'

(1) CurrentSCfile
Filename for current sensitivity coeffiecient values.

(type = character*80, default = CurrentSC.out)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
PrintCurrentSC MySensitivityCoefficients.dat
```

MESSAGEF*ILE – Filename for Event Message Output

Optional

The event messages are written to the file named on this record. If the record is not present, then the messages are written onto file MELMES_v2-0.

(1) MESSAGEFILE

Filename for special messages.

(type = character*80, default = MELMES_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
MESSAGEFILE BLOWDOWN_MES.TXT
```

STATUSF*ILE - Filename for MELCOR Status Request File

Optional

To allow the user to get information from a batch execution, MELCOR will check for the existence of a status file (using the FORTRAN inquire statement). If the status file exists, MELCOR will then generate and release a file named MELMAIL in the directory where the code is executing with a short execution status summary. The status file is then deleted. The default name is MELSTT_v2-0 (in the current directory). The file contents are unimportant because the file is never read. Also see STOPFILE to terminate batch execution.

(1) STATUSFILE

Filename for MELCOR status file.

(type = character*80, default = MELSTT_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
STATUSFILE BLOWDOWN_STAT
```

STOPF*ILE – Filename for MELCOR Stop File

Optional

To allow the user to terminate a batch execution, MELCOR will check for the existence of a stop file (using the FORTRAN inquire statement). If the stop file exists, then an edit, a plot dump, and a restart dump are generated, the calculation terminates, and the stop file is deleted. The default name of the stop file is MELSTP_v2-0. The file is never read, so the contents of the file are unimportant. See also STATUSFILE for control of batch executions.

(1) STOPFILE

Filename for MELCOR stop file.

(type = character*80, default = MELSTP_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
STOPFILE BLOWDOWN_STP
```

T_CON*TOUR – Filename for Temperature contour files

Optional

When the HTMLFILE is specified, HTML output is generated for the calculation, including certain graphics files. The name of the temperature contour files can be specified on this record. If this record is not present, temperature contour files will not be generated for the calculation. In order to view these files, certain javascript files must be available on your machine. The location of these javascript files is specified on the MEL_INSTALL record with the global variables.

(1) TCONFILE

Filename for HTML output.

(type = character*80, default = none)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
T_CONTOUR T_C_BLOWDOWN.HTML
```

WRITENEWINP – Write New Input File in Canonical Form

Optional

If this record is present, a file will be written containing input records corresponding to the state of the calculation at the point where MELCOR is started. The file is verbose and takes a very disciplined form. If it is written for CYCLE 0, it will describe the initial state.

(1) CANFIL

Filename to receive canonical input.

(type = character*80, default = MELGIN_v2-0.CAN)

If a file of this name already exists, the user will be prompted for permission to overwrite it.

Example

```
WRITENEWINP BLOWDOWN_MELGIN.CAN
```

EXTDIAGF*ILE – Filename for Extended Diagnostic Output File

Optional

The extended (unfiltered) diagnostic output from MELCOR is written to the file named on this record. If this record is not present, this output is written to extDIAG_v2-0.

(1) EXDIAF

Filename for extended diagnostic output.

(type = character*80, default = extDIAG_v2-0)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
EXTDIAGF BLOWDOWN_EXD
```

RN1VISUALFILE – Filename for RN Visualization Output File

Optional

Input to the RN package can lead to creation of optional special purpose output files to be used for visualization of aerosol and vapor inventories in selected volumes. A file will be created for each volume where each file name will contain the "root" name and the volume name. See the RN Users' Guide for more information.

(1) VISF

Root for filenames for visualization files.

(type = character*80, default = RN1VISDATA)

Note that the input processor by default converts lower case characters to upper case. If a lower case filename is desired, enclose the filename in single quotes.

Example

```
RN1VISUALFILE 'a&VV01aVIS'
```

NOTEPAD++ – Enables NotePad++ edititing characters in text output.

Optional

Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GPL License. This record enables special characters '(((' and ')))' which are used to generate a collapsible outline form in the text output. By default, no such characters will be added to the text output.

.

(1) On/OFF

'ON' (or no field) enables insertion of the special characters and 'OFF' disables it.

(type = character*80, default = 'ON')

Example

```
NOTEPAD++  ON
```

PrintInputRecords – Generates a complete list of input directives known to MELGEN.

Optional

A complete listing of all record identifiers known to MELGEN can be generated by inclusion of this record. The user provides the name of the text output file which will contain this listing. The listing is categorized by package and by type of record.

(1) RIDfilename

Provides the name of the record identifier output file.

(type = character*80, default = none)

Example

```
PrintInputRecords  MELGENrecords.dat
```

KEYWORDF*ILE – Generates a file containing plotvariable keywords formatted for ATLAS.

Optional

(1) Keyfilename

Provides the name of the key file.

(type = character*80, default = none)

Example

```
KeyWordF  ATLASkeys.dat
```

EXCELF*ILE – Currently unused.

4 MELGEN Input

During MELGEN execution, the EXEC package either prompts the user for an input filename or reads the filename as an execution line parameter.

MELGEN fname*

It is noted that this might not work with an executable compiled with the Digital FORTRAN compiler. All the input for the MELGEN execution is in this file. If the user responds to the input prompt with a carriage return only (blank filename), then MELGEN attempts to read input from the file MELGIN_v2-0. Only the EXEC_TITLE record is required, but the EXEC_DTTIME record is strongly recommended for use in MELGEN instead of MELCOR.

Like all input in the MELCOR suite of codes, each MELGEN input record has an identifier field followed by the input data fields, as discussed in Section 2. The MELGEN input records are listed below.

In the following, the 160-character limit should be used cautiously because only the first 160 characters of a record are recognized. As a result, fewer than 160 characters actually are available because of the characters required for the variable name and any spaces.

EXEC_INPUT – Start Record for EXEC User-input

Required

EXEC_DTTIME – Initial Timestep

Optional

This record defines the initial timestep for MELCOR execution. If this record is not present, then an initial timestep of 1 s is used. MELCOR will use the timestep fallback capabilities if the initial timestep is not appropriate for the transient.

(1) DT

Initial timestep.

(type = real, default = 1.0, units = s)

Example

```
EXEC_DTTIME    0.03 ! initial timestep
```

EXEC_JOBID – Job identifier

Optional

By default, each execution is assigned a unique computer-generated job identification. This record allows the user to override the default. The job identification appears in all output from the execution.

(1) JOBID

Job identifier field.

(type = character*9, default = unique character string)

Example

```
EXEC_JOBID 'HSPLT' ! job identifier
```

EXEC_PLOT – Control Function Argument to be Plotted

Optional

This record may be used to add elements of the MELCOR database that are available as control function arguments to the plot file. Multiple data input is required for arrays because each data string will add only a single variable. For example, plotting intact clad masses in multiple cells would require separate data strings for the control function argument COR-CellMassCL(IA,IR,MAT) for every cell (IA,IR) and material (MAT) of interest.

Variables added by EXEC_PLOT PLOTxxx input record in MELGEN will be written to the plot file throughout the entire MELCOR calculation. Further variables may be added using EXEC_PLOT input records in MELCOR.

(1) NUMVAR

Number of control function arguments to be plotted.

(type = integer, default = 0)

The following data are input as a table with length NUMVAR:

(1) NVAR

Data string index.

(type = integer, default = none, units = none)

(2) CHARG

Database element identifier (name of control function argument), including any parameters. Refer to the users' guides for the various packages for permitted values.

(type = character*160, default = none)

Example

```
EXEC_PLOT  4
     1 COR-VOL-FLU(7,1)
     2 COR-VOL-FLU(8,1)
     3 COR-VOL-FLU(9,1)
     4 COR-VOL-FLU(10,1)
```

EXEC_PLOTLENGTH – Fixed Length of Plot Key Name

Optional

For each plot variable sent to the binary plot file, an ASCII plot key name is also sent. For versions earlier than MELCOR 2.1, this plot key name was fixed at 24 characters. However, when adding extra plot variables to the plot file, this length may not be sufficient, particularly when adding control function arguments using the new MELCOR 2.1 format. MELCOR 2.1 will now vary the length of the plot key names to fit the longest plot key name, ensuring that this information is not truncated in the plot file. However, this variable length plot key may not be readable by early versions of PTFREAD (earlier than v1.7) as well as other postprocessors. This new input will restrict the length of the plot key header as specified by the user, even though this may mean truncating the plot key name, the intention being to make the plot file readable by existing postprocessors. If a plot key name exceeds the specified length, an error will be issued and the calculation will stop.

(1) NUMChars

Number of characters in the plot key name

(type = integer, default = 0)

Example

```
EXEC_PLOTLength  24  ! Enforces pre-MELCOR 2.1 plot key name length of 24 characters
```

EXEC_TITLE – Title of the Calculation

Required

Every calculation is required to have a title. This is written to the restart, edit, and plot files. The title on the MELCOR run must match a user-specified number of characters of the MELGEN title. If it contains blanks or if lower case characters are significant, the title must be enclosed in single quotes.

(1) TITLE

Title of the calculation.

(type = character*110)

Example

```
EXEC_TITLE  'DEMO, Version 2.0' ! Title of the calculation
```

EXEC_TSTART – Initial Start Time

Optional

The problem time may be set to a nonzero initial value. All package input must be consistent with the problem time.

(1) TSTART

Initial problem time. Negative times are permitted.

(type = real, default = 0.0, units = s)

Example

```
EXEC_TSTART    0.1 ! initial start time
```

EXEC_UNDEF – Redefine Initialization of Real Database and Scratch Storage

Optional

This capability is of interest primarily to code developers, but the information is provided in the interest of completeness.

The real database for MELCOR is explicitly initialized in MELGEN to a value UNDEF. The same is done with the scratch storage area before each advancement by each package in MELCOR. By default, a value is used that is both recognizable and likely to cause a code abort; the intent is to increase the probability of detecting coding errors that result in undefined variables. On some systems, this value is the so-called IEEE signaling *Not a Number* and causes an interrupt when used as an

operand in a floating point operation. On others, the value is -1.234E24. This input record allows the value of UNDEF to be changed, but not set to 0.0.

The initialization can be bypassed, if desired, although this is not recommended; the time savings are usually negligible.

(1) UNDEF

REAL value of UNDEF.

(type = real, default = machine dependent)
UNDEF='7fc00000'X on some machines using IEEE.
(default on HPW)
UNDEF='7f900000'X on some machines using IEEE.
(default on I60 and SUN)
UNDEF='80000000'X on VAXVMS *only*.
(default on VAXVMS)

or

(1) UNDOPT

CHARACTER string OFF to bypass the initialization.

(type = character*3, default = none, equivalence = none)

or

(1) UNDEF

CHARACTER string DEFAULT results in UNDEF = –1.234E24.

(type = character, default on other machines)

Example

```
EXEC_UNDEF    DEFAULT ! initial start time
```

EXEC_GLOBAL_DFT – Global Default Scheme Record

Optional

This record initially modifies a set of default and sensitivity coefficient values in all packages to those in the 2.x standards. The set of affected defaults and sensitivity coefficients are specified in each package UG under XXX_DFT record description. For example, affected COR package variables are listed under COR_DFT record description.

Note that these values may not reflect current MELCOR best practice modeling parameters and sensitivity coefficients for the analysis of severe accidents. If the record is present, one floating point field must be present. Permitted values are 1.86

for defaults to 1.8.6 standards and 2.0 for defaults to 2.x standards. Default value(s) of any particular record affected by this input can also be overridden by the user as in all other input records and sensitivity coefficients.

(1) DEFAULT

Mode of default scheme. Permitted values are 1.86 and 2.0

(type = real, default = 2.0, units = none)

Example

```
EXEC_GLOBAL_DFT 1.86 !Use 1.8.6 defaults for selected parameters
```

EXEC_SS – Use Accelerated Steady State

Optional

The problem may be started with an accelerated approach to steady state. This option works by using reduced heat capacities in the COR and HS structures. CV properties are unaltered. At the end of the steady state acceleration, internal energies and heat capacities are returned to normal.

(1) TSTART

Initial steady state start time. This is currently unused, and the steady state option starts at the problem start time.

(type = real, default = 0.0, units = s)

(2) TEND

End steady state acceleration time. The acceleration will end and heat capacities, etc. return to normal.

(type = real, default = 0.0, units = s)

(3) FACTR

Cp reduction factor.

(type = real, default = 0.01, units = none)

Example

```
EXEC_SS -5000.0 0.0 0.01 ! start SS at -5000s, resume normal at 0.0s
```

EXEC_WRT - Write temperature file for COR and HS temperatures

Optional

A file can be written containing input records in MELCOR input format to initialize the COR and HS temperatures in another related problem. The temperature file is in

MELCOR input format and can be included in another problem input with the same COR layout and HSs. This will also require using ALLOWREPLACE, probably commenting out the COR_CIT records in the including deck (the temperature file contains a COR_CIT record with a full set of temperatures for the core), and having the ISS flag on the HS_GD records set to NO (no steady state initialization) – otherwise, the temperature data on the node records in the temperature file will be ignored.

(1) TWRITE

The time at which to write the file with COR and HS temperatures.

(type = real, default = 0.0, units = s)

(2) TFILENAME (optional)

The filename of the output file

(type=character, default='Tifile.inp', units = none)

Example

```
EXEC_WRT 0.0! write file Tifile.inp at 0.0s
```

5 MELCOR Input

The MELCOR EXEC package either prompts the user for an input filename or reads the filename as an execution line parameter.

MELCOR fname

It is noted that this might not work with an executable compiled with the Digital FORTRAN compiler. All the input for the MELCOR execution is in this file. If the user responds to the input prompt with a carriage return only (blank filename), then MELCOR attempts to read input from file MELGIN_v2-0. If the input file cannot be opened, then the user will be asked if s/he wishes to try again. Only the EXEC_CPULEFT, EXEC_CPULIM, EXEC_TIME, and EXEC_TITLE records are required, but many of the remaining optional records are recommended for user convenience and flexibility.

Like all input in the MELCOR suite of codes, each MELCOR input record has an identifier field followed by the input data fields.

EXEC_INPUT – Start Record for EXEC MELCOR User-input

Required

EXEC_CFEXFILE – User specified CHANGEDATE File

Optional

This record specifies the file used to update values for READ and L-READ control functions. Values for the READ and L-READ control functions remain unchanged until MELCOR detects the presence of this file. After the file is read, it is deleted.

(1) CHANGEDATA

Name of file for updating READ and L-READ control functions.

(type = character*255, default = none, units = none)

(2) PAUSEFILE

Optional field specifying file that is used to pause MELCOR when it is placed in the working directory.

(type = character*255, default = none, units = none)

The format of records in the changedata file is:

(1) CFNAME

Name of READ or L-READ control function

(type = character*255, default = none, units = none)

(2) ReadValue

New value of control function. Type must be consistent with CF type for CFNAME.

(type = real or character*16, default = none, units = none)

(3) Comment

The comment following the (optional) '!' in the record is embedded in the message to allow identification of the change.

(type = real or character*256, default = none, units = none)

Example

```
EXEC_CFEXFILE CHANGEDATAFILE
Example (CHANGEDATEFILE Input)
CF_ID CF1 40.   !CF1 must be of type READ
CF_ID CF2 OFF   !CF2 must be of type L-READ
```

EXEC_COMTC – Number of Matching Characters in MELGEN and MELCOR Titles

Optional

Each MELGEN and MELCOR execution must have a title. A user–specified number of the characters on these titles must agree for the MELCOR calculation to proceed. If the titles do not agree with the user-specifed number of characters, then the calculation is stopped. This prevents the user from accidentally running the wrong

problem. It also allows the user to change the MELCOR runs, use the same MELGEN restart file, and change the end of the title to reflect the differences in the runs. COMTC must be greater than or equal to 20 and less than or equal to 80. If the input value is less than 20, then it is set to 20.

(1) NCOMTC

Number of characters to compare in the two titles.

(type = integer, default = 20, units = none)

Example

```
EXEC_COMTC    40
```

EXEC_CPULEFT – Desired Minimum Number of CPU Seconds Left at End of Calculation

Required

The calculation will stop after completing a cycle if the number of CPU seconds left in the calculation is less than the number input on this record. This record is used to stop the calculation with enough time to save files, generate plots, etc. This applies mainly to systems where jobs are submitted to a batch queue with a time limit. The description of the input MELCOR variable CPULIM discusses the method MELCOR uses in determining the number of CPU seconds remaining in a calculation.

(1) CPULEF

Desired minimum number of CPU seconds left at the end of the calculation. CPULEF must be non-negative.

- (type = real, default = none, units = s)

Example

```
EXEC_CPULEFT   20.0   ! cpu sec left at end of calculation
```

EXEC_CPULIM – Maximum Number of CPU Seconds Allowed for this Execution

Required

The maximum number of CPU seconds allowed for this execution is defined by the value for CPULIM input on this record. However, for some environments, a job CPU time limit external to MELCOR may also be imposed (for example, the maximum amount of job time defined on a batch submission utility). For some computer systems, EXEC is able to determine this limit using a system-dependent utility. When the CPU time used plus the safety factor CPULEF (input on record EXEC_CPULEFT) is greater than or equal to the minimum of CPULIM and the

externally imposed CPU limit, then a restart dump and edit will be written and the calculation will be terminated.

For interactive computer environments, CPULIM can be set to a large number and the calculation controlled by the interactive menu.

(1) CPULIM

Maximum number of CPU seconds allowed for this execution.

(type = real, default = none, units = s)

Example

```
EXEC_CPULIM 0.25E+05 ! max num of CPU sec allowed for execution
```

EXEC_CYMESF – Cycle Message Frequency Parameters

Optional

This record controls the frequency of the completed cycle messages written to the output and terminal files. In version 2.0, this record also controls the frequency with which MELCOR checks for external signals. One to four optional integer fields are allowed.

(1) NCYEDD

Number of cycles between messages written to the terminal or job stream file (unit 6). This can be reset from the interactive menu, as described in Section 10.

(type = integer, default = 10, units = none)

(2) NCYEDP

Number of cycles between messages written to OUTPUTFILE.

(type = integer, default = 1, units = none)

(3) NCYSTP

Number of cycles between checks for existence of a STOPFILE (default name MELSTP_v2-0, modifiable using the STOPF*ILE input record) or STATUSFILE (default name MELSTT_v2-0, modifiable using the STATUSF*ILE input record).

(type = integer, default = 1, units = none)

(4) NCYUIN

Number of cycles between checks for a possible user interrupt, as described in Section 10.

(type = integer, default = 1, units = none)

Example

```
EXEC_CYMESF     10   10   ! Cycle Message Frequency Parameters
```

EXEC_DTINCR – Timestep Increase Factor

Optional

MELCOR limits the increase in timestep between consecutive cycles to the ratio defined by this record. The default value allows an order of magnitude increase in timestep over five cycles.

(1) DTINCR

Maximum increase factor allowed in timestep.
Allowed range 1 < DTINCR < 2.
DT(next) ≤ DTINCR * DT(last).

(type = real, default = 101/5 = 1.5848932, units = none)

Example

```
EXEC_DTINCR   1.
```

EXEC_DTMAXCF – Time Step Limit Control Function

Optional

This record specifies the name of a real valued control function whose value (if greater than zero) will be imposed as an upper bound on the time step. This may be used to limit time steps during dynamic portions of a transient for which the relevant time period cannot be predicted in advance.

(1) IDTCF

Name of the REAL valued control function (which must have been defined as part of MELGEN input).

(type = character*16, default = none, units = none)

Example

```
EXEC_DTMAXCF   CF23
```

EXEC_DTSUMMARY – Additional Timestep Data to Output File

Optional

This record extends the edit of Δt information to the output file. One or two extra lines are produced for each cycle containing information on timestep requests by other packages. There is no additional input for this record.

Example

```
EXEC_DTSUMMARY
```

EXEC_DTTIME – Initial Timestep

Optional

This record defines the initial timestep for this execution. If this record is not present, then the initial timestep read from the restart file is used.

(1) DT

Initial timestep.

(type = real, default = from restart, units = s)

Normally, DTTIME should be input only in MELGEN and not in MELCOR. No differences will result when starting a new calculation at cycle = 0, but when restarting with cycle > 0, DTTIME will reset the timestep from what would be obtained in a single calculation without the restart, and results will be different.

Example

```
EXEC_DTTIME     0.1E-02   ! Initial timestep
```

EXEC_EDITCF – Special Edit Control Function

Optional

This record specifies the name of a LOGICAL valued control function that will generate an edit if its value is .TRUE. This is intended to force an edit at the time of some user-specified event. (If the control function is .FALSE., it will not prevent generation of an edit if requested by some other mechanism.)

(1) IECF

Name of the LOGICAL valued control function.

(type = character * 16, default = none, units = none)

Example

```
EXEC_EDITCF     CF245   ! Edit control function name
```

EXEC_EXACTTIME – Define Exact End-of-time-step Times

Optional

By default the timestep control does not attempt to hit any exact times in the determination of the values of Δt. The end-of-step time is simply the result of the summation of the Δts to that time. This record tells the timestep control to reduce the timestep shortly before the specified time so that the end-of-step time matches the desired time exactly. The intended use for this record is to allow certain time-specified events, such as reactor scram, pump trip, or valve closure, to be matched exactly, thus eliminating a potential source of numerical sensitivity. Any number of times may be defined, the values must be input in ascending order.

(1) NXTIME

Dimension of the exact times table.

(type = integer, default = 0, units = s)

The following data are input as a table with length NXTIME:

(1) N

Data string index.

(type = integer, default = none, units = s)

(2) TIME

Desired time.

(type = real, default = none, units = s)

Warning: This calculation will not override the minimum timestep DTMIN defined on the EXEC_TIME TIMEk input record. If multiple exact times are defined too close together, the later exact time will be ignored.

Example

```
EXEC_EXACTTIME   1 ! NUMBER  TIME
                       1     0.0
```

EXEC_FORCEPLOT – Control of Extra Plot Information Following Timestep Cuts

Optional

If there is a large timestep drop between consecutive cycles, MELCOR will generate plot dumps in addition to those defined by the EXEC_TIME record described in this document. This feature is designed to capture the features of rapid events that occur with unknown timing. This record allows the user to change the trigger for this feature and the maximum number of dump sequences generated.

(1) DT_RATIO

Ratio of consecutive cycle timesteps to trigger extra plot dumps.
If DT(i)/DT(i-1) < DT_RATIO, extra dumps are generated.
DT_RATIO must be less than one.

(type = real, default = 0.2, units = none)

(2) MAX_NUMBER_OF_SEQUENCES

Optional. Maximum number of extra plot dump sequences allowed per execution (i.e., the number of extra dump sequences generated is reset to zero when restarting a calculation).

(type = integer, default=20, units = none)

Example

```
EXEC_FORCEPLOT   0.3   25
```

EXEC_JOBID – Job Identifier

Optional

By default, each execution is assigned a unique computer-generated job identification. This record allows the user to override the default. The job identification appears in all output from the execution.

(1) JOBID

Job identifier field.

(type = character*9, default = unique character string)

Example

```
EXEC_JOBID   'SaltExp44'   ! Job identifier
```

EXEC_PLOT – Control Function Argument to be Plotted

Optional

This record may be used to add elements of the MELCOR database that are available as control function arguments to the plot file on restarts. Variables named on EXEC_PLOT input record in MELGEN will continue to be written to the plot file. Any number of variables may be added using EXEC_PLOT input record at a MELCOR restart and will be included in the plot file for the duration of the execution in which they are read.

(1) NUMVAR

Number of control function arguments to be plotted.

(type = integer, default = 0)

The following data are input as a table with length NUMVAR:

(1) NVAR

Data string index.

(type = integer, default = none, units = none)

(2) CHARG

Database element identifier (name of control function argument), including any parameters. Refer to the users' guides for the various packages for permitted values.

(type = character*160, default = none)

Example

```
EXEC_PLOT  3  ! Number             Name
                 1 RN1-ADEP(HS10001,LHS,CS,TOT)
                 2 RN1-ADEP(HS15001,RHS,CS,TOT)
                 3 RN1-ADEP(HS15002,RHS,CS,TOT)
```

EXEC_NOCOPY – Suppress Copy of Files to Output File

Optional

By default MELCOR will copy the message file (and diagnostic file in an error condition) to the end of the normal output file. This command suppresses the feature for those who prefer to keep these files separate.

Example

```
EXEC_NOCOPY
```

EXEC_NOFLUSH – Suppress Explicit Buffer Flushing

Optional

If this record (which contains no data fields) is present, there is no attempt to flush output buffers after data are written to the plot or restart file. The resulting behavior is processor-dependent and may involve loss of data when MELCOR terminates as a result of a system or arithmetic error.

Example

```
EXEC_NOFLUSH  ! Suppress Explicit Buffer Flushing
```

EXEC_PLOTCF – Special Plot Control Function

Optional

This record specifies the name of a logical valued control function that will generate a plot dump if its value is .TRUE. If the control function is .FALSE., then a plot dump will still be generated if some other mechanism (e.g., time, physics package request) requests it.

(1) IPCF

Name of a logical valued control function.

(type = character*16, default = none)

Example

```
EXEC_PLOTCF   CF77  ! Plot control function name
```

EXEC_RESTARTCF – Special Restart Control Function

Optional

This record specifies the name of a logical valued control function that will generate a restart dump if its value is .TRUE. If the control function is .FALSE., then a restart dump will still be generated if some other mechanism (e.g., time, physics package request) requests it.

(1) IRCF

Name of a logical valued control function.

(type = character*16, default = none, units = none)

Example

```
EXEC_RESTARTCF CF_RESTART ! Restart control function name
```

EXEC_SOFTDTMIN – Define Conditions for Limited Under-run of DTMIN Input

Optional

There are models in some MELCOR packages that require a major reduction in timestep at some event. An example is the High Pressure Melt Ejection, which may be activated at vessel failure. The exact timing of this event cannot be predicted by the user before running the calculation. The code requires a timestep at the start of ejection (e.g., $\Delta t \leq 10^{-4}$s) that is too small to use throughout the calculation. This input allows the user to accommodate this situation without restarting the execution and without specifying an excessively small DTMIN.

This input allows the timestep to be cut below the current DTMIN by a user-specified fraction for a limited number of times. Only cycles with Δt less than DTMIN and where Δt is decreasing from the previous cycle are counted. Δt less than DTMIN but increasing is allowed as many times as necessary to get within the user-specified range. See the EXEC_TIME input for user-specified limits.

(1) UNDERRUN_FRACTION

This is the fraction of DTMIN which is allowed. 10^{-2} to 10^{-3} are normally acceptable. Acceptable values are from 1 to 10^{-8}. The default condition does not allow under-runs.

(type = real, default = 0.01, units = none)

(2) NUMBER_OF_OCCURRENCES

Optional. This is the number of occurrences allowed per execution. The counter is reset to zero at restart.

(type = integer, default = 0, units = none)

Example

```
EXEC_SOFTDTMIN     0.1E-06
```

Consider the situation where the code was running with DTMIN=0.01 and Δt > DTMIN. The input record

```
EXEC_SOFTDTMIN 0.005 3
```

would allow the timestep to be suddenly dropped to as low as 5×10^{-5}s up to three times during the execution without terminating the run.

EXEC_STOPCF – Special Stop Control Function

Optional

This record specifies the name of a LOGICAL valued control function that will cause a graceful termination of the MELCOR calculation if its value is .TRUE. This function must have been defined as part of MELGEN input.

(1) ISCF

The name of a LOGICAL valued control function.

(type = character*16, default = none, units = none)

Example

```
EXEC_STOPCF   CF345 ! Stop control function name
```

EXEC_TIME – Timestep, Edit, Plot, and Restart Control

Required—at least one of table row required

The maximum and minimum timesteps, edit, plot, and restart frequency may be controlled using these records. The first input record determines when the remaining variables go into effect. Before that time the data on the previous record are used. After that time the data remain in effect until the problem time is greater than the time on the next record. The values on the last record remain in effect until the calculation finishes. MELCOR calculates its system timestep based on directives from the packages, but it cannot take timesteps greater than the maximum timestep or smaller than the minimum timestep. The maximum timestep must be at least a factor of two larger than the minimum. The edit, plot and restart dump frequencies are controlled by these records. If a user specifies an edit frequency of 120 seconds, then every 120 seconds a new edit is generated. Any of the following can be changed from the interactive menu:

(1) NTMINV
Dimension of the timesteps table.
(type = integer, default = none, units = none)

The following data are input as a table with length NTMINV:

(1) N
Data string index.
(type = integer, default = none, units = none)

(2) TIME
Time the data on this record go into effect.
(type = real, default = none, units = s)

(3) DTMAX
Maximum timestep allowed during time interval.
(type = real, default = none, units = s)

(4) DTMIN
Minimum timestep allowed during time interval.
(type = real, default = none, units = s)

(5) DTEDIT
Edit frequency during this time interval.
(type = real, default = none, units = s)

(6) DTPLOT
Plot frequency during this time interval.
(type = real, default = none, units = s)

(7) DTREST

Restart frequency during this time interval.

(type = real, default = none, units = s)

(8) DCREST

Restart frequency based on CPU time during this time interval. This field is optional.

(type = real, default = 1010, units = s, minimum=100.0)

Example

```
EXEC_TIME 4
!N TIME     DTMAX   DTMIN   DTEDT   DTPLT   DTRST    DCRST
 1 0.00E+00 0.1E+01 0.1E-02 0.1E+04 0.2E+01 0.50E+02 0.1E+11
 2 0.45E+03 0.1E+01 0.1E-02 0.1E+04 0.2E+01 0.50E+03 0.1E+11
 3 0.10E+04 0.1E+01 0.1E-02 0.1E+04 0.2E+01 0.15E+05 0.1E+11
 4 0.20E+04 0.1E+01 0.1E-02 0.1E+04 0.2E+01 0.15E+05 0.1E+11
```

EXEC_TEND – End of Calculation Time

Optional

The calculation will stop when the problem time is greater than or equal to TEND. The calculation may also terminate because the CPU time is exhausted, interactive termination is requested, or one or more packages request the calculation be stopped. A restart file is written at the last cycle before the calculation terminates. This variable can be reset from the interactive menu, as discussed in Section 10.

(1) TEND

End of calculation time.

(type = real, default = 5.4321 x 1020, units = s)

Example

```
EXEC_TEND    0.1495E+05 ! end of calculation time
```

EXEC_TITLE – Title of the Calculation

Required

Every calculation is required to have a title. This is written to the restart, edit, and plot files. The title on the MELCOR run must match a user-specified number of characters of the MELGEN title. See the COMTC command. If it contains blanks or if lower case characters are significant, the title must be enclosed in single quotes.

(1) TITLE

Title of the calculation.

(type = character*80)

Example

```
EXEC_TITLE  'DEMO, Version 1.8.5' * title of the calculation
```

EXEC_UNDEF – Redefine Initialization of Real Database and Scratch Storage

Optional

This capability is of interest primarily to code developers, but the information is provided in the interest of completeness.

The real database for MELCOR is explicitly initialized in MELGEN to a value UNDEF. The same is done with the scratch storage area before each advancement by each package in MELCOR. By default, a value is used that is both recognizable and likely to cause a code abort; the intent is to increase the probability of detecting coding errors that result in undefined variables. On some systems, this value is the so-called IEEE signaling *Not a number* and causes an interrupt when used as an operand in a floating point operation. On others, the value is -1.234E24. This input record allows the value of UNDEF to be changed but not set to 0.0.

The initialization can be bypassed, if desired, although this is not recommended; the time savings are usually negligible.

(1) UNDEF

REAL value of UNDEF.

(type = real, default = machine dependent)
UNDEF='7fc00000'X on some machines using IEEE.
(default on HPW)
UNDEF='7f900000'X on some machines using IEEE.
(default on I60 and SUN)
UNDEF='80000000'X on VAXVMS *only*.
(default on VAXVMS)

or

(1) UNDOPT

CHARACTER string OFF to bypass the initialization.

(type = character*3, default = none, equivalence = none)

or

(1) UNDEF

CHARACTER string DEFAULT results in UNDEF = –1.234E24.

(type = character, default on other machines)

Example

```
EXEC_UNDEF   OFF
```

EXEC_WARNINGL*EVEL – Define Level for Warning and Information Messages

Optional

MELCOR can generate many forms of warning and information messages during an execution. These messages go to the output, terminal, and/or diagnostic files. In some cases, the volume of messages can be a problem. This record allows the user to control the volume and severity of the messages. Each message is assigned a level in the range 0 to 5. A fatal error is assigned the level 0. The least important messages (to the user in a successful execution) are assigned the level 5, and other messages are assigned intermediate levels according to relative importance. Messages with levels 4 and 5 are normally of interest only in the location of a code problem. This record tells the code to only record messages with a level equal to or less than the values input.

This feature applies only to the time advancement (run) mode of MELCOR. Messages generated during the setup mode (MELGEN) are always listed.

The range of each of the following is 0 to 5 except for terminal output. The terminal parameter allows an additional input of -1 to suppress messages from the message package being displayed on the terminal. This was the situation before MELCOR 1.8.2.

(1) MLEVO

Output file message level.

(type = integer, default = 2, units = none)

(2) MLEVD

Optional. Diagnostic file message level. If input field is not present, then level is set to the output file level.

(type = integer, default = 2, units = none)

(3) MLEVT

Optional. Terminal message level. If input field is not present, then level is set to the output file level.

(type = integer, default = -1, units = none)

Note: Many of the MELCOR packages have not yet been modified to utilize this feature. The modifications will be included as rapidly as possible.

Example

```
EXEC_WARNINGL   3   3
```

6 MELCOR Sensitivity Coefficient Modification

A goal for MELCOR is to implement as sensitivity coefficients all model or correlation constants and parameters, convergence criteria, and similar system parameters that are not normally accessible via input. These sensitivity coefficients are grouped into numbered arrays that represent a set of similar parameters, such as the several constants appearing in a single correlation. These arrays are identified by a four-digit number NNNN incorporated into the array names. Each MELCOR package is assigned a range of identifier numbers for the sensitivity coefficients in that package. The various sensitivity coefficients are described in the individual package users' guides.

These sensitivity coefficients can be redefined in MELGEN or changed at any restart via MELCOR input (rather than modification of any code) so that the effects of modeling changes on the calculation can be examined. Each XYZ package (having sensitivity coefficients) has its own subroutines for processing any input that modifies the package sensitivity coefficients. The value of any sensitivity coefficient is changed by using an input record of the following form:

XYZ_SC – XYZ Sensitivity Coefficients Input Record

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index, used for identifying and ordering the input.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the XYZ sensitivity coefficient array.
(type = integer, default = none, units = none)

(3) VALUE
New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.
(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA
First index of the sensitivity coefficient.
(type = integer, default = none, units = none)

(5) NB
Second index of the sensitivity coefficient (if required).
(type = integer, default = none, units = none)

The number of indices required will depend on the sensitivity coefficient. Data string indices NSTR used for identifying and ordering the table rows may follow one after another in arbitrary sequence. The only requirement is that they must be unique integer numbers. Note, only one XYZ_SC table with sensitivity coefficients input data may be used for each XYZ package. This means that input data for sensitivity coefficients with different identifiers NNNN must be placed in one combined table one by one.

For example, the following sensitivity coefficient input will change the default minimum oxidation cutoff temperature and Zircaloy-steel eutectic temperature of the Core (COR) package from the values of 1100 K and 1400 K to 1200 K and 1450 K respectively:

```
COR_SC   2  ! N   Identifier   Value       Index
              1   1004         1.2E+3        1 ! Minimum temp
              2   1011         1.45E+3       2 ! Zircaloy-steel
```

EXEC does not have any sensitivity coefficients.

7 Plot Variables

The elements of the EXEC database that may be used for plot variables are listed below, along with a brief description. Note that control function arguments (some that are identical in definition to the plot variables but different in format) are described in the following section.

WARP	Problem time minus initial problem time, divided by CPU time. (units = dimensionless)
LOCALWARP	Same as WARP, except computed with decaying time history (i.e., calculated over a few cycles and weighted more heavily for more recent cycles) to indicate current performance. (units = dimensionless)

8 Control Function Arguments

The elements of the EXEC database that may be used for control function arguments are listed below, along with a brief description. Note that plot variables (some that are

identical in definition to these control function arguments but different in format) are described in the previous section.

EXEC-TIME	Problem time. (units = s)
EXEC-DT	Timestep. (units = s)
EXEC-CYCLE	Cycle number. (units = dimensionless)
EXEC-CPU	CPU time used. (units = s)
EXEC-WARP	Problem time minus initial problem time, divided by CPU time. (units = dimensionless)
EXEC-LOCALWARP	Same as WARP, except computed with decaying time history (i.e., calculated over a few cycles and weighted more heavily for more recent cycles) to indicate current performance. (units = dimensionless)

9 MELGEN and MELCOR Execution

In version 2.0 (F95) all the names of used files are listed in a separate, ENVIRONMENTAL block of the input deck; MELGEN and MELCOR execution use this and only this information about interaction with the file system. MELGEN and MELCOR can be executed with or without command line parameters. In the most general form, the permitted parameters are

```
i=input_file_name
sf=stop_file_name
ow=x
```

Any or all of these may be present, in any order. None may contain embedded blanks.

The character string following "i=" is interpreted as the name of the input file (default MELGIN or MELIN), and that following "sf=" as the name of the stop file (ordinarily specified on input record STOPF*ILE, with default M

In ow=x, x specifies the action that is to be taken if the name of a text output file to be written duplicates the name of a file that already exists in the working directory. "ow=o" allows any and all such files to be overwritten; "ow=e" preserves the contents of such

files, with new output simply appended to them; "ow=a" (or "ow=x" where "x" is anything except "o" or "e") will cause MELCOR to simply abort if any such file exists.

The input file name can also be specified *without the "i="* as either the only parameter on the command line or the first parameter. If there are no parameters, the code will prompt for the name of the input file.

The following forms are acceptable for MELGEN execution (if melgen is replaced by melcor, they are acceptable for MELCOR execution):

(1) melgen

(2) melgen filename

(3) melgen sf=stopper i=filename

(4) melgen ow=o i=filename

(5) melgen filename ow=o

In case (1), the code will prompt for the input filename. In all cases except (4) and (5), the code will prompt for instructions if the name of a text output file duplicates the name of a file already existing in the working directory; in case (4) and (5), any such file will be overwritten without warning. In case (3), the presence of a file named "stopper" in the working directory will terminate a MELCOR execution, even if input record STOPF*ILE is included in the input file and specifies a different name.

Additional MELCOR commands can be added to the end of the command line. These commands can be used to over-ride default value or values specified in the input deck. For example, the ending time may be specified on the command line. If specified, this value replaces any value that is specified in the input. Other commands include the following:

C=	Specification of enabled named comment blocks (e.g., c=case1+case2+case3)
DTMAX=	Over-ride calculation maximum time step [sec]
N++=	Generate output textfile that is formatted for Notepad++ outlining ('ON' or 'OFF'
NCF=	Enable/Disable new critical flow model ('ON' or 'OFF')
NQE=	Disable/enable not-quite equilibrium model ('ON' or 'OFF')
NT=	Specify number of threads for multi-thread version.
OW=	Overwrite/Extend/Abort option
SCnnnn(m)=	Specify value of sensitivity coefficient SCnnnn(m)
SF=	Specify name of stop file

ST=	Specify matrix solver (1=DSDBCG, 2=PARDISO, 3=UMFPACK, 4=DSDBCGSTAB
TEND=	Over-ride calculation end time [sec]
VAR=	User specification of variable input values, (e.g., VAR={{{VAR1=0.05}}}+{{{VAR2=10.5}}})

Example

```
MELGEN.exe  input.inp  ow=e TEND=10000.0 SC4401(1)=0.0005
```

10 MELCOR Interactive Interrupts

Interactive executions of MELCOR can be interrupted by certain messages typed at the terminal. Processed messages are

STATUS! or ! for one-line status message
HELLO! for interactive menu
STOP!, QUIT!, END!, or EXIT! to terminate execution
OFF! to turn off interactive input (required for some debuggers to function correctly)

Commands are case-insensitive. Any character string not on this list will produce a brief list of recognized commands.

The interactive menu options are somewhat limited at the present time and will be expanded in future releases. Current options include

(1) List DT data including statistical information.

(2) Terminate calculation at end of cycle.

(3) Interactive full edit (standard long edit with editor).

(4) Reset stop time.

(5) Write full edit to separate file and release.

(6) Reset terminal cycle message frequency.

(7) Suspend execution for specified time.

(8) Reset CPU limit.

(9) Reset time interval input for time step minimum and maximum limits, edit interval, plot interval, etc.

(A) Activate tracing of the performance of the CVH package, as described for the CVHTRACE record in the CVH Package Users' Guide.

(X) Force immediate generation of an additional restart dump, plot point, and/or text edit to the appropriate file(s).

After the interactive options are completed, MELCOR returns to normal execution.

11 Example Input

The following example input illustrates the use of several of the MELGEN and MELCOR input records.

```
! Comments common to MELGEN and MELCOR
!
MEG_DIAGFILE    tmidiagg
MEL_DIAGFILE    tmidiag
MEG_OUTPUTFILE  tmioutg
MEL_OUTPUTFILE  tmiout
MEG_RESTARTFILE tmirest
MEL_RESTARTFILE tmirest    NCYCLE     0
PLOTFILE        tmiplot.ptf
MESSAGEFILE     tmimes
Program MELGEN
!* Block: EXEC data ************************************
EXEC_INPUT
EXEC_TITLE  "TMI-2" ! Title of the calculation
EXEC_RUNONLY NM
EXEC_DTTIME  0.1
EXEC_PLOT  3
     1 COR-CellMassCL(3,1,ZR)
     1 COR-CellMassCL(4,1,ZR)
     1 COR-CellMassCL(12,1,ZR)
!* END EXEC ********************************************
INCLUDE cvhinput
INCLUDE corinput
INCLUDE hsinput
INCLUDE rninput
INCLUDE otherinput
...
END Program MELGEN data
!
Program MELCOR
!* Block: MEX (Exec) data ****************************
EXEC_INPUT
EXEC_TITLE   TMI-2' ! Title of the calculation
EXEC_RUNONLY NM
EXEC_TIME  2 !  NUMBER TIME DTMAX DTMIN DTEDT DTPLT DTRST DCRST
                 1    0.0   5.0 0.0001 500.0  2.0 500.0  0.1E+11
                 2  1000.0 10.0 0.0001 1000.0 5.0 1000.0 0.1E+11
EXEC_CPULEFT   30.0
EXEC_CPULIM    36000.0
EXEC_TEND      6000.0
EXEC_NOCOPY
EXEC_PLOT  3
     1 RN1-ADEP(HS30001,LHS,CS,TOT)
     2 RN1-ADEP(HS30001,LHS,I2,TOT)
     3 RN1-ADEP(HS30001,LHS,CSI,TOT)
```

```
!* END MEX (Exec) ****************************
INCLUDE scinput
END Program MELCOR data
```

Fan Cooler (FCL) Package
Users' Guide

The MELCOR Engineered Safety Features (ESF) package models the phenomena for the various engineered safety features in a nuclear power plant. The Fan Cooler (FCL) package constitutes a subpackage of the ESF package and calculates the heat and mass transfer associated with operation of the fan coolers. This users' guide provides basic information required to run the FCL model package with the rest of MELCOR, including a detailed explanation of the user input and package output for MELGEN, MELCOR, and post processors. Required and optional input, sensitivity coefficients, control function arguments, plot variables, and error messages are all explained.

More detailed information on the phenomenological modeling and numerical solution schemes implemented in the FCL package can be found in the FCL Package Reference Manual.

Contents

1 INTRODUCTION 3

2 INPUT REQUIREMENTS 3

2.1 MELGEN User Input 3

FCL_ID – Fan Cooler Name 3
FCL_ICI – Fan Cooler Interface and Control 5
FCL_RFT – Fan Cooler Rated Flows and Temperatures 5
FCL_ARC – Additional Fan Cooler Rated Conditions 6
FCL_AFT – Fan Cooler Actual Flows and Temperatures 6

2.2 MELCOR Input 7

3 SENSITIVITY COEFFICIENTS 7

3.1 FCL Sensitivity Coefficients 7

9001 – Coefficients for MARCH Fan Cooler Heat Transfer Correlation 7
ESF_SC – ESF Sensitivity Coefficients Input Record 8

4 PLOT VARIABLES 9

5 CONTROL FUNCTION ARGUMENTS 9

6 EXAMPLE INPUT 10

7 FCL MODEL OUTPUT 10

8 DIAGNOSTICS AND ERROR MESSAGES 10

APPENDIX 1 11

1 Introduction

The MELCOR fan cooler (FCL) model package is based on the fan cooler model in the MARCH 2.0 code. An effective heat transfer area is calculated in MELGEN from the rated primary and secondary flows and temperatures, and from the heat transfer coefficient and cooler capacity at those conditions. The actual heat transfer rate during a transient is calculated based on the effective area, the heat transfer coefficient for the current steam mole fraction, and the average temperatures of the primary gas and secondary coolant, which are implicit functions of the heat transfer rate. Details of the model can be found in the FCL Package Reference Manual.

Several extensions to the model have been made. The user may optionally specify a separate discharge control volume for the fan cooler outlet airflow. The user may also specify a control function to control operation of the cooler by turning it on or off. Finally, the MELCOR implementation roughly partitions the total heat transfer coefficient into separate convection and condensation components in an attempt to account for the effects of noncondensible gases and superheated atmosphere. The user can control how this partitioning is made by adjusting the sensitivity coefficients used in the heat transfer correlation.

2 Input Requirements

This section describes the input requirements for the MELCOR FCL package, including a short description of the input quantities and their units and default values, if any. Further description of the input variables and their meaning in the models can be found in the FCL Package Reference Manual.

Input for the FCL model must be within the data block for the ESF package and begins with the character string ESF_INPUT. In addition to the simple fan cooler model based on the MARCH modeling, a more mechanistic model based on the mechanistic model that was part of the CONTAIN code has been implemented. Multiple fan coolers can be specified, and input is grouped into sets for each fan cooler modeled, identified by record FCL_ID.

2.1 MELGEN User Input

FCL_ID – Fan Cooler Name

Required

This record specifies a user-supplied name for the fan cooler. This record is required. The following character field (limited to 16 characters) must be present:

(1) FCNAME

Fan cooler name.

(character*16, default = none)

(2) FCMODEL

Fan Cooler Model to be used

"MECHANISTIC" or "CONTAIN" - mechanistic fan cooler model

"MARCH"– old MARCH modeling

(character*16, default = "MARCH")

Example

```
FCL_ID  Fan1  CONTAIN
```

FCL_HT – Heat Transfer Parameters for Mechanistic Model

Optional

This record specifies the heat transfer parameters for the fan cooler. This record is optional but should be specified with parameters characteristic of the user's design.

(1) NCOILS

Number of fan cooler rows from front to back of cooler.

(type = integer, default = 12, units = none)

(2) DCOIL

Outer diameter of fan cooler coil.

(type = real, default = 0.0159, units = m)

(3) AreaHT

Effective heat transfer area for one row of coils. Because tubes have closely spaced fins, this value is several times the tube area.

(type = real, default = 26.5 m^2, units = m^2)

(4) AreaFL

Frontal flow area of cooler

(type = real, default = 3.0 m^2, units = m^2)

(4) HTCEFF

Effective heat transfer coefficient through the boundary layer and coil

(type = real, default = 1000.0 m^2, units = W/m^2-K)

Example

```
FCL_HT  12   0.0159      26.5   3.0    1000.0
```

FCL_ICI – Fan Cooler Interface and Control

Required

This record specifies the control volumes to which the fan cooler is interfaced and a control function to turn the cooler on or off. This record is required, but only the first field must be present; the remaining two fields are optional.

(1) ICVI

Fan cooler inlet control volume name.

(type = character*16, default = none, units = none)

(2) ICVD

Fan cooler discharge control volume name. If this field is omitted, the discharge volume is the same as the inlet volume, i.e., the fan cooler sits entirely within a control volume. If this field is different from the inlet volume, the fan cooler operates somewhat like a flow path with a constant volumetric flow (that is cooled or dehumidified) from the inlet volume to the discharge volume.

(type = character*16, default = ICVI, units = none)

(3) ICF

The name of the fan cooler logical control function that determines whether the fan cooler is on or off. This control function should return a value of .TRUE. whenever the fan cooler should be turned on. If this field is omitted, the fan cooler is always on.

(type = character*16, default = none, units = none)

Example

```
FCL_ICI  CVol100       CVol100        CF20
```

FCL_RFT – Fan Cooler Rated Flows and Temperatures

Required

This record specifies the rated primary and secondary flow rates and inlet temperatures. This record is required. NOTE: Care must be exercised to ensure that rated flows and temperatures are consistent with the rated cooler capacity.

(1) XVFGSR

Rated fan cooler gas volumetric flow rate.

(type = real, default = none, units = m^3/s)

(2) XMFSER

Rated fan cooler secondary coolant mass flow rate.

(type = real, default = none, units = kg/s)

(3) TSECIR

Rated fan cooler secondary coolant inlet temperature.

(type = real, default = none, units = K)

(4) TPR

Rated fan cooler inlet gas temperature.

(type = real, default = none, units = K)

Example

```
FCL_RFT  100.0       65.0        294.0       339.0
```

FCL_ARC – Additional Fan Cooler Rated Conditions

Required

This record specifies additional fan cooler rated conditions. This record is required. NOTE: Care must be exercised to ensure that rated flows and temperatures are consistent with the rated cooler capacity.

(1) QRAT

Rated fan cooler capacity.

(type = real, default = none, units = W)

(2) FMLSTR

Steam mole fraction at rated conditions.

(type = real, default = none, units = none)

Example

```
FCL_ARC  1.9E6          0.693
```

FCL_AFT – Fan Cooler Actual Flows and Temperatures

Optional

This record specifies the actual primary volumetric flow rate, secondary mass flow rate, and inlet temperature during the transient if different from the rated values. If zero or a negative number is input for a field, it defaults to the rated value. This record is optional, but if any of the fields require changes from default values, the entire set of three variables must be supplied.

(1) XVFGSI

Actual fan cooler gas volumetric flow rate.

(type = real, default = XVFGSR, units = m^3/s)

(2) XMFSEC

Actual fan cooler secondary coolant mass flow rate.

(type = real, default = XMFSER, units = kg/s)

(3) TSECIN

Actual fan cooler secondary coolant inlet temperature.

(type = real, default = TSECIR, units = K)

Example

```
FCL_AFT  -1.        -1.         314.0
```

2.2 MELCOR Input

No input for the fan cooler model is processed during MELCOR execution.

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. Their use is described in Section 6 of the MELCOR Executive (EXEC) Users' Guide.

3.1 FCL Sensitivity Coefficients

This section lists the sensitivity coefficients in the FCL model that are accessible to the user, along with a brief description, and gives their default values, units, and EQUIVALENCE names.

9001 – Coefficients for MARCH Fan Cooler Heat Transfer Correlation

These coefficients are used to calculate the effective heat transfer coefficient (W/m^2-K), as a function of the steam mole fraction X_{H2O}. The total heat (energy) transferred results from sensible and latent heat transfers. The sensible heat portion of the heat transfer coefficient, h_H, is calculated by

$$h_H = C9001(1) \cdot C9001(2)$$

The latent heat portion of the heat transfer coefficient, h_M, is calculated by

$$h_M = h_L + C9001(1) \cdot [1 - C9001(2)]$$

where

$$h_L = C9001(4) \cdot X_{H2O}, X_{H2O} \le C9001(3)$$

$$h_L = C9001(3) \cdot C9001(4) + C9001(5)[X_{H2O} - C9001(3)], \; X_{H2O} > C9001(3)$$

The default values are taken from the MARCH correlation.

(1) Sensible heat transfer coefficient.
(default = 590.54, units = W/m^2-K, equiv = HSEN)

(2) Sensible heat transfer multiplier.
(default = 1.0, units = none, equiv = FSEN)

(3) Steam mole fraction (boundary) in latent heat transfer coefficient correlation.
(default = 0.26, units = none, equiv = FMLSCR)

(4) Coefficient in latent heat transfer coefficient correlation for low steam mole fractions.
(default = 3603.4, units = W/m^2-K, equiv = DHLAT1)

(5) Coefficient in latent heat transfer coefficient correlation for high steam mole fractions.
(default = 2325.25, units = W/m^2-K, equiv = DHLAT2)

ESF_SC – ESF Sensitivity Coefficients Input Record

Optional

This record allows the user to change the default ESF sensitivity coefficient values to user-defined inputs. This record is not required.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Table record index.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the ESF sensitivity coefficient array.
(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
ESF_SC   2  ! N   SC          VALUE     INDEX
              1   9001        0.3           3
              2   9001        2500.0        5
```

4 Plot Variables

The plot variables are described in this section. Note that control function arguments (some are identical in definition to these plot variables but different in format) are described in the following section.

ESF-QFC-RAT.n	Heat transfer rate for fan cooler index n. (units = W)
ESF-QFC-TOT.n	Total energy transfer for fan cooler index n. (units = J)
ESF-MFC-RAT.n	Condensation rate for fan cooler index n. (units = kg/s)
ESF-MFC-TOT.n	Total steam condensed for fan cooler index n. (units = kg)

5 Control Function Arguments

The control function arguments are described in this section. Note that plot variables (some are identical in definition to these control function arguments but different in format) are described in the previous section.

ESF-QFC-RAT(NameFCL)	Heat transfer rate for fan cooler name NameFCL. (units = W)
ESF-QFC-TOT(NameFCL)	Total energy transfer for fan cooler name NameFCL. (units = J)

ESF-MFC-RAT(NameFCL)	Condensation rate for fan cooler name NameFCL. (units = kg/s)
ESF-MFC-TOT(NameFCL)	Total steam condensed for fan cooler name NameFCL. (units = kg)

6 Example Input

Sample MELGEN input records for the FCL model for a fan cooler treated by the MARCH-based model are included in this section. No MELCOR input records are necessary to run the FCL model.

```
ESF_INPUT
...
FCL_ID  FCL1
!       ICVI       ICVD      ICF
FCL_ICI CV100      CV100   'ON100'

!          XVFGSR     XMFSER  TSECIR    TPR
FCL_RFT     100.0      65.0    294.0   339.0

!       QRAT       FMLSTR
FCL_ARC 1.9E6       0.693
*
!         XVFGSI XMFSEC TSECIN
FCL_AFT    -1.     -1.   314.0
...
CF_INPUT
...
! CONTROL FUNCTION FOR FAN COOLER
! TURN FAN COOLER ON WHEN TIME GT 100.
CF_ID  'ON100' L-GT
CF_LIV FALSE
CF_ARG 2 ! NARG CHARG         ARSCAL   ARADCN
             1 EXEC-TIME       1.0       0.0
             2 EXEC-TIME       0.0      100.0
```

7 FCL Model Output

In general, the FCL model output is self-explanatory. The heat transfer and condensation rates, the total cumulative energy transferred, and steam condensed are output for each fan cooler.

8 Diagnostics and Error Messages

Diagnostics and error messages generated during MELGEN relate to input processing and are generally self-explanatory. Currently, no messages are generated during MELCOR execution.

Appendix 1

Melcor 2.1	PTFRead	Comments
ESF-QFC-RAT(NameFCL) NameFCL – the fan cooler name.	ESF-QFC-RAT.n n – the fan cooler index.	Heat transfer rate for fan cooler name NameFCL (index n).
ESF-QFC-TOT(NameFCL) NameFCL – the fan cooler name.	ESF-QFC-TOT.n n – the fan cooler index.	Total energy transfer for fan cooler name NameFCL (index n).
ESF-MFC-RAT(NameFCL) NameFCL – the fan cooler name.	ESF-MFC-RAT.n n – the fan cooler index.	Condensation rate for fan cooler name NameFCL (index n).
ESF-MFC-TOT(NameFCL) NameFCL – the fan cooler name.	ESF-MFC-TOT.n n – the fan cooler index.	Total steam condensed for fan cooler name NameFCL (index n).

Fuel Dispersal Interactions (FDI) Package Users' Guide

The MELCOR Fuel Dispersal Interactions (FDI) package models both low-pressure molten fuel ejection from the reactor pressure vessel (RPV) into the reactor cavity and high-pressure molten fuel ejection from the RPV with the possibility of dispersion of the debris over multiple containment volumes and surfaces. The possibility of steam explosions from fuel-coolant interactions is not considered. In MELCOR 1.8.5, a FDI sensitivity coefficient is used to control the numerical order in which oxygen or steam is used to oxidize direct containment heating (DCH) metals. This parameter affects the amount of hydrogen that results from burning DCH materials in steam/oxygen atmospheres.

This document includes a brief description of the models employed, the input format, sample input, sensitivity coefficients, plot variables, and control functions. Details on the models can be found in the FDI Package Reference Manual.

Contents

1 INTRODUCTION 3

2 USER INPUT 4

2.1 MELGEN User Input 5

FDI_ID – FDI Name 5
FDI_LOC – FDI Location and Transfer Process Names 5
FDI_EL – Elevations 9
FDI_CVH – HPME Control Volume Parameters 9
FDI_DEP – HPME Deposition Surface Parameters 11

2.2 MELCOR User Input 13

FDI_ID – FDI Name 13
FDI_CVH – HPME Control Volume Parameters 13
FDI_DEP – HPME Deposition Surface Parameters 15

3 SENSITIVITY COEFFICIENTS 17

3.1 FDI Sensitivity Coefficients 17

4602 – High/Low Ejection Velocity Transition 17
4603 – Airborne Debris Temperature Limit 17
4604 – Maximum Change in CVH Atmosphere Temperature per Timestep 17
4605 – Pool Water Ejection Ratio 18
4606 – Minimum Airborne Mass Ratio 18
4607 – Initial Timestep Size for HPME Initiation 18
4608 – Maximum Debris-to-Wall Heat Transfer Coefficient 18
4609 – Minimum Debris Temperature for Oxidation 18
4610 – Oxygen/Steam Oxidation Weighting Factor 19
4620 – Convergence Criteria for the FDI Equation of State 19
FDI_SC – FDI Sensitivity Coefficients Input Record 19

4 PLOT VARIABLES 20

5 CONTROL FUNCTION ARGUMENTS 26

6 EXAMPLE INPUT 30

6.1 Settling Time Constant 33

6.2 Heat Transfer Time Constant 33

6.3 Oxidation Time Constant 34

APPENDIX 1 35

1 Introduction

The capabilities of the Fuel Dispersal Interactions (FDI) package are briefly summarized in this section. More details will be provided in the FDI Package Reference Manual.

Two types of phenomena are treated in the FDI package:

(1) Low-pressure molten fuel ejection (LPME) from the reactor vessel, and

(2) High-pressure molten fuel ejection (HPME) from the reactor vessel, (direct containment heating [DCH]).

There is currently no plan to model steam explosions within or outside the FDI package in MELCOR.

During low-pressure ejection, heat is transferred from the molten fuel to the water pool (if present in the associated control volume) as it breaks up and falls to the cavity floor. The heat transfer normally occurs by radiation, but a convective lower bound is also included. All of the energy transfer from the molten fuel is used to boil the pool water (sensible heating is assumed to be unimportant). If no pool is present, material passes through FDI without any energy removal.

If the velocity of the molten debris ejected from the reactor vessel exceeds a critical value, prescribed by an adjustable sensitivity coefficient, or if the user has invoked the stand-alone option for high-pressure melt ejection modeling, then the fuel dispersal interactions will be treated by the high-pressure model instead of the low-pressure model. The parametric high-pressure model requires user input to control both the distribution of debris throughout the containment and the interaction of the hot debris with the containment atmosphere and heat structure (deposition) surfaces. The processes modeled include oxidation of the metallic components of the debris (zircaloy, aluminum, and steel are considered) in both steam and oxygen, surface deposition of the airborne debris by trapping or settling, and heat transfer to the atmosphere and deposition surfaces. First-order rate equations with user-defined time constants for oxidation, heat transfer, and settling are used to determine the rate of each process. Heat transfer to structure surfaces is limited by a heat transfer coefficient specified by a sensitivity coefficient. Debris entering the Cavity (CAV) package, either by direct deposition or settling from the atmosphere, is not treated by the FDI package; hence, the oxidation and heat transfer on deposition surfaces refers only to heat structure deposition surfaces. If a pool of water exists in the reactor cavity at the time of debris ejection, then the model ejects the water into the droplet field (fog) of the atmosphere at a rate proportional to the rate of injection of the debris into the pool.

If HPME model user input is absent, the HPME model will be disabled, and all FDI events will be treated by the LPME model irrespective of ejection conditions. No DCH will occur if the user has not included the necessary input on records FDI_CVH and FDI_DEP.

2 User Input

The user input for the FDI package is described in this section. MELGEN input is described first, followed by MELCOR input. The FDI model may be applied at several melt ejection locations within a problem; however, in the vast majority of problems the model is applied at only one location. An FDI location is basically a location at which debris may be ejected from a reactor vessel upon failure or a location at which debris is introduced from a source external to the problem domain. In most full-scale reactor calculations the model is applied in the Control Volume Hydrodynamics (CVH) control volume beneath the reactor vessel lower head (specified by entry ICVCAV on Core (COR) record COR_LHD). In stand-alone HPME analyses of DCH issues, the model is applied in the volume immediately downstream of the debris source.

For reactor plant calculations that may lead to vessel failure at relatively high pressure, users should provide input for both the LPME and HPME models. The LPME model input designates the FDI location, which is normally the reactor cavity CVH volume name as discussed above, the transfer process name associated with debris transfer from the COR package to the FDI package, the CAV cavity name, and the associated transfer process name for debris transferred from the FDI package to the CAV package. The HPME model input specifies volumes into which the debris may be injected and parameters for controlling the interaction between the debris and its surroundings. If HPME input is not provided, the LPME model will be invoked irrespective of the differential pressure between the reactor vessel and the reactor cavity during the melt ejection.

The FDI model may also be used to explore DCH issues and experiments which do not involve actual modeling of the reactor (COR package). In such cases, the HPME model is used in stand-alone mode, and the high-pressure debris source to the model is provided by the user. In stand-alone HPME calculations, the LPME model is inactive, and the LPME model input that normally controls the LPME model is used to control the stand-alone HPME model.

The stand-alone HPME model is invoked by setting NFDCAV = -1 on record FDI_LOC. For the stand-alone HPME model, there are two options for entering the debris source. The first option is to enter the debris source via the External Data File (EDF) package (see EDF package documentation). The second option is to enter the debris source via tabular function input (see Tabular Function (TF) package documentation). For the second option, the debris source temperature is input via tabular function (name input as TFNAM on record FDI_LOC), and the name of materials sourced in with separate tabular functions (functions identified on FDI_LOC records using ITABLE fields) must be input.

The COR package must not be active when using the stand-alone HPME model.

Only the parameters that control the distribution of HPME debris throughout containment and the parameters that control the interaction of HPME debris with the containment atmosphere can be changed on restart in the MELCOR input. Therefore, it is important for users to determine which control volumes and deposition surfaces will be included in the

database at the time MELGEN is executed, because volumes and surfaces cannot be added later.

2.1 MELGEN User Input

One set of the following records is required for each FDI location. Input records FDI_LOC and FDI_EL are required to activate either the LPME or HPME model, and in addition, the FDI_CVH and FDI_DEP records are required to activate the HPME model.

FDI_ID – FDI Name

Required

(1) FDINAM

Name of FDI location.

(type = character*16, default = none, units = none)

Example

FDI_ID FDI2

FDI_LOC – FDI Location and Transfer Process Names

Required

This record identifies the control volume, cavity, *in* transfer process name (for transfers to CORCON), and *out* transfer process name (for transfers from COR) for each FDI location. NOTE: The stand-alone HPME model applications are activated by setting NFDCAV = -1.

(1) NFDICV

User-defined name of associated control volume.

(type = character*16, default = none, units = none)

(2) NFDCAV

Key, designating one of the following values.

(a) 0 or CAV
Use cavity.

(b) -1 or NOCAV
Do not use CAV and COR packages; the stand-alone HPME model is invoked.

(type = integer / character*5, default = none, units = none)

If NFDCAV = 0 or CAV, the following additional fields are required:

(3) CAVNAM

User-defined name of associated cavity.

(type = character*16, default = none, units = none)

(4) TPINAM

In transfer process name for transfers to CORCON.

(type = character*16, default = none, units = none)

(5) TPONAM

Out transfer process name for transfers from COR or EDF.

(type = character*16, default = none, units = none)

(6) RNTPINAM

In transfer process name for RadioNuclide (RN1) package.

(type = character*16, default = none, units = none)

(7) RNTPONAM

Out transfer process name for RN1 package.

Required if RN1 package is active.

(type = character*16, default = none, units = none)

If NFDCAV = -1 or NOCAV, the following additional fields are required:

(3) DEBRSRC

Options for entering the debris source.

(a) 1 or TF

Enter the debris source via tabular function input (see TF package documentation).

(b) -1 or TPOUT

Enter the debris source via the Transfer Process (TP) package (see EDF package documentation).

(type = integer/character, default = none, units = none)

If DEBRSRC = 1 or TF, the following two fields and the table are required:

(4) TFNAM

Tabular function name for the table of source temperature versus time.

(type = character*16, default = none, units = none)

The following fields describe the materials sourced into the stand-alone HPME model by tabular function input. The tabular function should be the total mass of the specified material sourced in to the specified table time (integral of the mass source rate).

(5) NUMTF

Number of tabular functions for material source and dimension of the table below.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMTF. The data are input from the new line.

(1) NMAT

Source material index used for sequencing.

(type = integer, default = none, units = none)

(2) MATNAM

Source material name.

Valid Entries for MATNAM:

(a) ZIRCALOY
(b) ZIRCONIUM-OXIDE
(c) URANIUM-DIOXIDE
(d) STAINLESS-STEEL
(e) STAINLESS-STEEL-OXIDE
(f) BORON-CARBIDE
(g) SILVER-INDIUM-CADMIUM
(h) URANIUM-METAL
(i) ALUMINIUM
(j) ALUMINIUM-OXIDE
(k) CADMIUM

(NOTE: Entries are case insensitive.)

(type = character*24, default = none, units = none)

(3) ITABLE

Tabular function name for material source MATNAM.

(type = character*16, default = none, units = none)

If DEBRSRC = -1 or TPOUT, then

(4) TPONAM

Out transfer process name for transfers from EDF (an external data file is used for the stand-alone HPME source).

(type = character*16, default = none, units = none)

(5) RNTPONAM

Out transfer process name for RN1 package. Required if RN1 package is active.

(type = character*16, default = none, units = none)

Material is transferred from the COR package (or an EDF file in the stand-alone HPME application) to the FDI package and from the FDI package to the CAV package through the TP package. The user must define transfer processes (in the TP package input) to perform the transfers. For transferring material from COR to FDI, an *in* transfer process must be defined to transfer material into the TP package from the COR package and input (as NTPCOR) on the COR_TP record in the COR package input. A corresponding *out* transfer process, with name TPONAM (above), must be defined to remove material from the TP package and transfer it into the FDI package. Transfers between FDI and CAV are performed similarly. They involve the *in* process TPINAM (above) and an *out* process input as NTPOT on the CAV_TP record in the CAV package input. If desired, material may be transferred directly from the COR package to the CAV package, bypassing the FDI package.

2.1 FDI Location and Transfer Process Names

FDI location:				
1. LPME or HPME model				
COR	→ NTPCOR $\xrightarrow{MTX}$ TPONAM →	FDI	→ TPINAM $\xrightarrow{MTX}$ NTPOT →	CAV
2. Standalone HPME model				
TF	TFNAM (temperature) + NMAT x ITABLE (materials) →	FDI		
EDF	→ TPIN $\xrightarrow{MTX}$ TPONAM →	FDI		

For transfers from FDI to CAV, the field TYPE for the corresponding *in* process (NAMETPIN = TPINAM) on the TP_IN record must be FDI. The number of masses and the number of thermodynamic variables are in the order defined by the package NMSIN = 5 and NTHRM = 9 (see Table 3.1 of the TP Package Users' Guide). The TP_MTX record for the corresponding TPONAM should define NROW = 9 and NCOL = 5. The order of masses ejected from the FDI package is:

(1) UO_2,
(2) Zr,
(3) Steel,
(4) ZrO_2, and
(5) Steel oxide.

Additional discussion of FDI-CAV transfers is included in the CAV Users' Guide.

Note: In order to transfer the radionuclides along with the fuel and metal masses, additional transfer process names must be defined. Currently, these radionuclide transfer process names must be input in addition to the corresponding transfer process names defined for mass and energy transfers among the COR, FDI, and CAV packages. The transfer process input should set the number of masses equal to the total number of radionuclide classes, NTHRM equals to 1, and should specify a unity translation matrix (DEF1 for the corresponding *out* process on the TP_OUT record).

Example

```
FDI_LOC CVolume732 NOCAV TF TF613 5
                                 !NMAT     MATNAM         ITABLE
                                  1  ZIRCALOY              TF634
                                  2  ZIRCONIUM-OXIDE       TF635
                                  3  URANIUM-DIOXIDE       TF636
                                  4  STAINLESS-STEEL       TF637
                                  5  STAINLESS-STEEL-OXIDE TF638
FDI_LOC CV732      CAV   CAV1 TP102 TP101
```

FDI_EL – Elevations

Required

This record specifies the elevation at which material begins undergoing the FDI (this will usually be the elevation of the bottom head penetration) and the elevation of the floor on which the material is deposited after the FDI is completed.

(1) ZBOTTM

Floor elevation. Must lie within the control volume identified by NFDICV on record FDI_LOC. If not coincident with the bottom of that volume, a warning message will be issued.

(type = real, default = none, units = m)

(2) ZTOP

Top elevation of interaction region. Must be greater than ZBOTTM. If above the top of the associated control volume, a warning message will be issued.

(type = real, default = none, units = m)

Example

```
FDI_EL    6.04    7.3
```

<u>High-Pressure Model Input:</u>

Records FDI_CVH and FDI_DEP must only appear together. The inclusion of these records activates the HPME model. The sum of FATM and FDEP over all the volumes and surfaces associated with each FDI location must equal one, or an error message will be issued.

FDI_CVH – HPME Control Volume Parameters

Required if NFDCAV = -1 or NOCAV, otherwise, optional

(1) NATM

Number of control volumes HPME debris enters for this FDI location, dimension of the following table.

(type = integer, default = none, units = none)

The following data are input as a table with length NATM. The data are input from the new line.

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) IDATM

User-defined name of CVH volume receiving debris.

(type = character*16, default = none, units = none)

(3) IACTV

Options that indicate type of surface.

(a) 1 or CAV
For CORCON cavity.

(b) 2 or LHS
For left-hand side of Heat Structures (HS) package structure.

(c) 3 or RHS
For right-hand side of HS structure.

(type = integer / character*3, default = none, units = none)

If IACTV = CAV, then

(4a) NAMECAV

User-defined name of CAV cavity debris that will settle from volume IDATM. (This must be one of the surfaces listed on FDI_DEP input records.)

Required if IACTV = 1 or CAV only.

(type = character*16, default = none, units = none)

else, i.e., IACTV = LHS or RHS, then

(4b) NAMEHS

User-defined name of HS surface where debris will settle from volume IDATM. (This must be one of the surfaces listed on FDI_DEP input records.)

Required if IACTV = 2 or LHS, or if IACTV = 3 or RHS.

(type = character*16, default = none, units = none)

Whatever the case, the following fields are required:

(5) FATM

Fraction of debris ejected from this location that enters atmosphere of control volume name IDATM.

(type = real, default = none, units = none)

(6) TOXV

Time constant for oxidation reactions in the atmosphere of control volume name IDATM. The time constant will have a value equal to the absolute value of TOXV. A positive value for TOXV indicates that a hierarchical scheme will be used in which the order of oxidation is Zr, Al and then steel. A negative value of TOXV indicates that oxidation of all metals will occur simultaneously.

(type = real, default = none, units = s)

(7) THT

Time constant for heat transfer to the atmosphere of control volume name IDATM.

(type = real, default = none, units = s)

(8) TST

Time constant for settling onto surface IACTV with name NAMEHS or NAMECAV from control volume name IDATM.

(type = real, default = none, units = s)

NOTE: Time constants with absolute values smaller than 10^{-6} s will be reset to 10^{-6} s to avoid numerical problems.

Examples

```
FDI_CVH 1 ! N IDATM ITYP CAV/HS    FATM    TOXV    THT    TST
            1 CV732 CAV  CAV2       0.9     1.5    1.5    0.2
FDI_CVH 2 ! N    IDATM  ITYP    CAV/HS    FATM  TOXV  THT  TST
            1 CVolume712  2   HeatSlab703 0.45  0.8   0.3  0.2
            2 CVolume732 LHS  HeatSlab701 0.45  0.7   0.3  0.2
```

FDI_DEP – HPME Deposition Surface Parameters

Required if NFDCAV = -1 or NOCAV, otherwise, optional

This record describes the surfaces on which debris is deposited from both settling (from control volumes) and/or direct deposition from the HPME source.

(1) NDEP

Number of deposition surfaces associated with this FDI location by settling or direct deposition from the HPME source, dimension of the table below.

(type = integer, default = none, units = none)

The following data are input as a table with length NDEP:

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) ITYP

Options which indicate what type of surface IDDEP is

(a) 1 or CAV
For CORCON cavity.

(b) 2 or LHS
For left-hand side of HS structure.

(c) 3 or RHS
For right-hand side of HS structure.

(type = integer / character*3, default = none, units = none)

If ITYP = LHS or RHS, then

(3) IDDEP

User-defined name of HS surface that receives debris from settling and/or direct deposition from the HPME source.

(type = character*16, default = none, units = none)

(4) FDEP

Fraction of debris ejected from this location which is deposited directly on surface IDDEP (independent of settling from control volumes).

(type = real, default = none, units = none)

(5) TOXS

Time constant for oxidation reactions on surface IDDEP. The time constant will have a value equal to the absolute value of TOXS.

A positive value of TOXS indicates that a hierarchical scheme will be used in which the order of oxidation is Zr, Al and then steel. A negative value of TOXS indicates that oxidation of all metals will occur simultaneously.

(type = real, default = none, units = s)

Otherwise, if ITYP = CAV, then

(3) IDDEP

User-defined name of CAV cavity that receives debris from settling and/or direct deposition from the HPME source.

(type = character*16, default = none, units = none)

(4) FDEP

Fraction of debris ejected from this location which is deposited directly on surface IDDEP (independent of settling from control volumes).

(type = real, default = none, units = none)

(5) NFDITP

In transfer process name associated with the deposition process to a CORCON cavity.

(type = character*16, default = none, units = none)

(6) RNTPINAM

In transfer process name associated with the RN1 package. Required if RN1 package is active.

(type = character*16, default = none, units = none)

Example

```
FDI_DEP 2 ! N ITYP IDDEP  FDEP   TOXS / NFDITP RNTPINAM
            1 CAV  CAV2   0.03          TP602
            2 LHS  HS702  0.07   1.5
```

2.2 MELCOR User Input

Only control volumes and deposition surfaces that were initially defined in the database may be modified so that the values of NATM and NDEP do not exceed their MELGEN values. The sum of FATM and FDEP over all volumes and surfaces associated with FDI location FDINAM must be equal to one, or an error message will be issued.

FDI_ID – FDI Name

Required if MELCOR input for this FDI location is defined

(1) FDINAM

Name of FDI location.

(type = character*16, default = none, units = none)

FDI_CVH – HPME Control Volume Parameters

Optional

(1) NATM

Number of modified control volumes. It must be less than the number of control volumes that were initially defined in MELGEN, dimension of the table below.

(type = integer, default = none, units = none)

The following data are input as a table with length NATM. The data are input from the new line.

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) IDATM

User-defined name of modified CVH volume receiving debris (it must be among the control volume names defined in MELGEN).

(type = character*16, default = none, units = none)

(3) ITYP

Options that indicate surface type.

(a) 1 or CAV
For CORCON cavity.

(b) 2 or LHS
For left-hand side of HS structure.

(c) 3 or RHS
For right-hand side of HS structure.

(type = integer / character*3, default = none, units = none)

If ITYP = CAV, then

(4a) NAMECAV

User-defined name of CAV cavity where debris will settle from volume IDATM.

Required if ITYP = 1 or CAV.

(type = character*16, default = none, units = none)

else, i.e., ITYP = LHS or RHS, then

(4b) NAMEHS

User-defined name of HS surface where debris will settle from volume IDATM.

Required if ITYP = 2 or LHS, or if ITYP = 3 or RHS.

(type = character*16, default = none, units = none)

In any case, the following fields are required:

(5) FATM

Fraction of debris ejected from this location that enters atmosphere of control volume name IDATM.

(type = real, default = none, units = none)

(6) TOXV

Time constant for oxidation reactions in the atmosphere of control volume name IDATM. The time constant will have a value equal to the absolute value of TOXV. A positive value for TOXV indicates that a hierarchical scheme will be used in which the order of oxidation is Zr, Al and then steel. A negative value of TOXV indicates that oxidation of all metals will occur simultaneously.

(type = real, default = none, units = s)

(7) THT

Time constant for heat transfer to the atmosphere of control volume name IDATM.

(type = real, default = none, units = s)

(8) TST

Time constant for settling onto surface IACTV with name NAMEHS or NAMECAV from control volume name IDATM.

(type = real, default = none, units = s)

NOTE: Time constants with absolute values smaller than 10^{-6} s will be reset to 10^{-6} s to avoid numerical problems.

Example

```
FDI_CVH 1 ! N IDATM ITYP CAV/HS    FATM    TOXV    THT    TST
            1 CV732 CAV  CAV2      0.9     1.5     1.5    0.2
```

FDI_DEP – HPME Deposition Surface Parameters

Optional

This record describes the modified surfaces on which debris is deposited from both settling (from control volumes) and/or direct deposition from the HPME source.

(1) NDEP

Number of modified deposition surfaces associated with this FDI location by settling or direct deposition from the HPME source. It must be less than number of deposition surfaces that were initially defined in MELGEN, dimension of the table below.

(type = integer, default = none, units = none)

The following data are input as a table with length NDEP:

(1) N

Table record index.

(type = integer, default = none, units = none)

(2) ITYP

Options that indicate what type of surface IDDEP is

(a) 1 or CAV
For CORCON cavity.

(b) 2 or LHS
For left-hand side of HS structure.

(c) 3 or RHS
For right-hand side of HS structure.

(type = integer / character*3, default = none, units = none)

If ITYP = LHS or RHS, then

(3) IDDEP

User-defined name of HS surface that receives debris from settling and/or direct deposition from the HPME source (it must be among the control deposition surfaces defined in MELGEN).

(type = character*16, default = none, units = none)

(4) FDEP

Fraction of debris ejected from this location that is deposited directly on surface IDDEP (independent of settling from control volumes).

(type = real, default = none, units = none)

(5) TOXS

Time constant for oxidation reactions on surface IDDEP. The time constant will have a value equal to the absolute value of TOXS.

A positive value of TOXS indicates that a hierarchical scheme will be used in which the order of oxidation is Zr, Al and then steel. A negative value of TOXS indicates that oxidation of all metals will occur simultaneously.

(type = real, default = none, units = s)

If ITYP = CAV, then

(3) IDDEP

User-defined name ofCAV cavity that receives debris from settling and/or direct deposition from the HPME source.

(type = character*16, default = none, units = none)

(4) FDEP

Fraction of debris ejected from this location that is deposited directly on surface IDDEP (independent of settling from control volumes).

(type = real, default = none, units = none)

Example

```
FDI_DEP 1 ! N ITYP IDDEP  FDEP   TOXS / NFDITP RNTPINAM
            1 LHS  HS702  0.08   1.6
```

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. Their use is described in Section 7 of the MELCOR EXEC Users' Guide.

3.1 FDI Sensitivity Coefficients

4602 – High/Low Ejection Velocity Transition

Vessel ejection velocity at transition between high- and low-pressure ejection modeling. The ejection velocity calculated in the COR package is compared to this coefficient to determine whether to invoke the high- or low-pressure model.

(default = 10, units = m/s, equiv = none)

4603 – Airborne Debris Temperature Limit

If the airborne debris temperature exceeds this value, then any oxidation energy is deposited directly into the atmosphere to simulate the effect of a rapid increase in debris-to-gas heat transfer caused by debris fragmentation associated with rapidly escalating internal vapor pressure above the debris boiling point.

(default = 3700, units = K, equiv = none)

4604 – Maximum Change in CVH Atmosphere Temperature per Timestep

If DCH will change the CVH atmosphere temperature by more than this value in a single timestep, then the FDI package will request a timestep fallback.

(default = 500, units = K, equiv = none)

4605 – Pool Water Ejection Ratio

Proportionality constant between the mass of water ejected from the cavity pool and the mass of debris injected into the pool from the HPME source. If X kg of debris is ejected from the COR package in a timestep, then X*C4605(1) kg of pool water is transferred to the fog component of the reactor cavity atmosphere during that timestep.

(default = 10, units = none, equiv = none)

4606 – Minimum Airborne Mass Ratio

The ratio of the current airborne mass to the integrated airborne debris mass source in a control volume below which the mass will be deposited on the settling surface associated with the control volume—deactivates DCH when the airborne mass remaining in a control volume becomes negligible.

(default = 0.001, units = none, equiv = none)

4607 – Initial Timestep Size for HPME Initiation

If the current timestep size exceeds this value at HPME model initiation, then the FDI package will request a timestep fallback.

(default = 0.0001, units = s, equiv = DTHPME)

4608 – Maximum Debris-to-Wall Heat Transfer Coefficient

The rate of heat transfer from deposited debris to the deposition surface is determined by a heat transfer time constant which is equal to MIN(0.5*TOXS,.001). The intention is to establish equilibrium between the debris temperature and the surface temperature on a reasonably short time scale. However, as the amount of deposited debris accumulates, a very large heat flux to the surface can result. This sensitivity coefficient limits the rate of heat transfer if that value exceeds $HTC_{MAX} * A_{SRF}\left(T_{OLD} - T_{SRF}\right)$, where HTC_{MAX} is the value of sensitivity coefficient 4608, A_{SRF} is the surface area of the deposition surface, T_{OLD} is the temperature of the deposited debris at the beginning of the timestep, and T_{SRF} is the deposition surface temperature (from the HS package database).

(default = 1000, units = W/m^2-K, equiv = HTCMAX)

4609 – Minimum Debris Temperature for Oxidation

The temperature of the debris (either airborne or deposited) must exceed this value, or oxidation of the metallic components of the debris is not permitted.

(default = 600, units = K, equiv = TOXMIN1)

4610 – Oxygen/Steam Oxidation Weighting Factor

Weighting factor to control relative oxidation of debris by oxygen versus steam. The default value of 0.5 will give equal weighting proportional to their relative mole fractions in the atmosphere. A value of 0.0 will result in oxidation using all available steam in preference to oxygen, and a value of 1.0 will result in oxidation by available oxygen in preference to steam.

(default = 0.5, units = none, equiv = WGTO2)

4620 – Convergence Criteria for the FDI Equation of State

These coefficients are used to specify convergence tolerances for determining a temperature mass and enthalpy. In general, convergence is assumed if the enthalpy is matched with a relative error less than TOLENH. In cases where the state corresponds to the melting point of one of the constituent materials, convergence is also assumed if the temperature is bounded in a range with relative width TOLTMP.

(1) Tolerance (relative) for enthalpy.
(default = 1.0E-6, units = dimensionless, equiv = TOLENH)

(2) Tolerance (relative) for temperature.
(default = 2.0E-7, units = dimensionless, equiv = TOLTMP)

FDI_SC – FDI Sensitivity Coefficients Input Record

Optional

This record allows for change of the default FDI sensitivity coefficient values by the user. This record is not required.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the FDI sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example:

```
FDI_SC   4  ! N   SCnumber     Value       Index
              1   4602         15.0         11
              2   4603         4.0E3        15
              3   4620         1.5D-6        2
              4   4620         5.0D-7        3
```

4 Plot Variables

The FDI package variables that may be used for plot variables are described below. Note that control function arguments (some are identical in definition to these plot variables but different in format) are described in the following section. In the following list n refers to the FDI location index, m refers to the control volume index associated through MELGEN input with FDI location n, and k refers to the heat structure index associated through MELGEN input with FDI location n.

FDI-FMREL.n	Mass of material released to CORCON from FDI location n for this timestep. (kg)
FDI-FMRELT.n	Integrated mass released to CORCON from FDI location n for entire calculation. (kg)
FDI-ETRAN.n	Energy transferred to water from FDI location n for this timestep. (J)
FDI-ETRANT.n	Integrated energy transferred to water from FDI location n for entire calculation.

	(J)
FDI-STGEN.n	Mass of steam generated in FDI location n for this timestep. (kg)
FDI-STGENT.n	Integrated steam mass generation for FDI location n for entire calculation. (kg)
FDI-ZR-OXRAT.n.m	Zircaloy oxidation rate in CVH volume m for FDI location n. (kg/s)
FDI-ZR-OXTOT.n.m	Integral over time of the zircaloy oxidation rate in CVH volume m for FDI location n. (kg)
FDI-AL-OXRAT.n.m	Aluminum oxidation rate in CVH volume m for FDI location n. (kg/s)
FDI-AL-OXTOT.n.m	Integral over time of the aluminum oxidation rate in CVH volume m for FDI location n. (kg)
FDI-SS-OXRAT.n.m	Steel oxidation rate in CVH volume m for FDI location n. (kg/s)
FDI-SS-OXTOT.n.m	Integral over time of the steel oxidation rate in CVH volume m for FDI location n. (kg)
FD-O2-OXRAT.n.m	Oxygen consumption rate in CVH volume m for FDI location n. (kg/s)
FDI-O2-OXTOT.n.m	Integral over time of the oxygen consumption rate in CVH volume m for FDI location n. (kg)
FDI-ST-OXRAT.n.m	Steam consumption rate in CVH volume m for FDI location n. (kg/s)

FDI-ST-OXTOT.n.m	Integral over time of the steam consumption rate in CVH volume m for FDI location n. (kg)
FDI-H2-OXRAT.n.m	Hydrogen generation rate in CVH volume m for FDI location n. (kg/s)
FDI-H2-OXTOT.n.m	Integral over time of the hydrogen generation rate in CVH volume m for FDI location n. (kg)
FDI-ATM-POWR.n.m	Heat transfer rate to atmosphere in CVH volume m for FDI location n. (W)
FDI-ATM-HEAT.n.m	Heat transferred to atmosphere in CVH volume m for FDI location n. (J)
FDI-DEBRIS-T.n.m	Airborne debris temperature in CVH volume m for FDI location n. (k)
FDI-OX-ENRGY.n.m	Energy generated by the oxidation of zircaloy and steel in CVH volume m for FDI location n. (J)
FDI-MASS-ADD.n.m	Mass transferred from TP package to CVH volume m for FDI location n. (kg)
FDI-ENTH-ADD.n.m	Enthalpy transferred from TP package to CVH volume m for FDI location n. (J)
FDI-ATM-ZRM.n.m	Airborne mass of zircaloy in CVH volume m for FDI location n. (kg)
FDI-ATM-ZRX.n.m	Airborne mass of zirconium oxide in CVH volume m for FDI location n. (kg)
FDI-ATM-UO2.n.m	Airborne mass of uranium dioxide in CVH volume m for FDI location n.

	(kg)
FDU-ATM-SSM.n.m	Airborne mass of steel in CVH volume m for FDI location n. (kg)
FDI-ATM-SSX.n.m	Airborne mass of steel oxide in CVH volume m for FDI location n. (kg)
FDI-ATM-ALM.n.m	Airborne mass of aluminum in CVH volume m for FDI location n. (kg)
FDI-ATM-ALX.n.m	Airborne mass of aluminum oxide in CVH volume m for FDI location n. (kg)
FDI-ATM-UMT.n.m	Airborne mass of uranium metal in CVH volume m for FDI location n. (kg)
FDI-ATM-CDM.n.m	Airborne mass of cadmium in CVH volume m for FDI location n. (kg)
FDI-ATM-B4C.n.m	Airborne mass of boron carbide in CVH volume m for FDI location n. (kg)
FDI-ATM-AIC.n.m	Airborne mass of silver-indium-cadmium in CVH volume m for FDI location n. (kg)
FDI-ATM-ALL.n.m	Airborne mass of all components in CVH volume m for FDI location n. (kg)
FDI-ZR-SXRAT.s.n.k	Zircaloy oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-ZR-SXTOT.s.n.k	Integral over time of the zircaloy oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)

FDI-AL-SXRAT.s.n.k	Aluminum oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-AL-SXTOT.s.n.k	Integral over time of the aluminum oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SS-SXRAT.s.n.k	Steel oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-SS-SXTOT.s.n.k	Integral over time of the steel oxidation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-O2-SXRAT.s.n.k	Oxygen consumption rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-O2-SXTOT.s.n.k	Integral over time of the oxygen consumption rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-ST-SXRAT.s.n.k	Steam consumption rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-ST-SXTOT.s.n.k	Integral over time of the steam consumption rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-H2-SXRAT.s.n.k	Hydrogen generation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg/s)
FDI-H2-SXTOT.s.n.k	Integral over time of the hydrogen generation rate on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)

FDI-SRF-POWR.s.n.k	Heat transfer rate to the s (LHS or RHS) side of HS structure k for FDI location n. (W)
FDI-SRF-HEAT.s.n.k	Heat transferred to the s (LHS or RHS) side of HS structure k for FDI location n. (J)
FDI-TBD-SURF.s.n.k	Temperature of deposited debris on the s (LHS or RHS) side of HS structure k for FDI location n. (K)
FDI-SX-ENRGY.s.n.k	Energy generated by the oxidation of metals on the s (LHS or RHS) side of HS structure k for FDI location n. (J)
FDI-MASS-SET.s.n.k	Total mass that has settled out of the atmosphere onto the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-ENTH-SET.s.n.k	Total enthalpy that has settled out of the atmosphere onto the s (LHS or RHS) side of HS structure k for FDI location n. (J)
FDI-SRF-ZRM.s.n.k	Mass of zircaloy on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-ZRX.s.n.k	Mass of zirconium oxide on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-UO2.s.n.k	Mass of uranium dioxide on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-SSM.s.n.k	Mass of steel on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-SSX.s.n.k	Mass of steel oxide on the s (LHS or RHS) side of HS structure k for FDI location n.

	(kg)
FDI-SRF-ALM.sn.k	Mass of aluminum on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-ALX.s.n.k	Mass of aluminum oxide on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-UMT.s.n.k	Mass of uranium metal on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-CDM.s.n.k	Mass of cadmium on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-B4C.s.n.k	Mass of boron carbide on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-AIC.s.n.k	Mass of silver-indium-cadmium on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)
FDI-SRF-ALL.s.n.k	Mass of all components on the s (LHS or RHS) side of HS structure k for FDI location n. (kg)

5 Control Function Arguments

The FDI package variables that may be used for control function arguments are listed and described below. Note that plot variables (some are identical in definition to these control function arguments but different in format) are described in the previous section. In the following list FDIName refers to the FDI location name, CVHName refers to the CVH user volume names associated through MELGEN input with FDI location FDIName, and HSName refers to the HS user heat structure names associated through MELGEN input with FDI location.

FDI-FMREL(FDIName)	Mass of material released to CORCON from FDI location FDIname for this timestep. (kg)

FDI-FMRELT(FDIName)

Integrated mass released to CORCON from FDI location FDIname for entire calculation.

(kg)

FDI-ETRAN(FDIName)

Energy transferred to water from FDI location FDIname for this timestep.

(J)

FDI-ETRANT(FDIName)

Integrated energy transferred to water from FDI location FDIname for entire calculation.

(J)

FDI-STGEN(FDIName)

Mass of steam generated in FDI location FDIname for this timestep.

(kg)

FDI-STGENT(FDIName)

Integrated steam mass generation for FDI location FDIname for entire calculation.

(kg)

FDI-OXRAT(FDIName,CVHName,MatName)

Oxidation/consumption/generation rate of material MatName in CVH volume CVHName for FDI location FDIName. The choices allowed for MatName are: ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen.

(kg/s)

FDI-OXTOT(FDIName,CVHName,MatName)

Integral over time of the oxidation/consumption/ generation rate of the material (MatName) in CVH volume CVHName for FDI location FDIName. The choices allowed for MatName are: ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen.

(kg)

FDI-ATM-POWR(FDIName,CVHName)

Heat transfer rate to atmosphere in CVH volume CVHName for FDI location FDIName.

(W)

FDI-ATM-HEAT(FDIName,CVHName) Heat transferred to atmosphere in CVH volume CVHName for FDI location FDIName.
(J)

FDI-DEBRIS-T(FDIName,CVHName) Airborne debris temperature in CVH volume CVHName for FDI location FDIName.
(k)

FDI-OX-ENRGY(FDIName,CVHName) Energy generated by the oxidation of zircaloy and steel in CVH volume CVHName for FDI location FDIName.
(J)

FDI-MASS-ADD(FDIName,CVHName) Mass transferred from TP package to CVH volume CVHName for FDI location FDIName.
(kg)

FDI-ENTH-ADD(FDIName,CVHName) Enthalpy transferred from TP package to CVH volume CVHName for FDI location FDIName.
(J)

FDI-ATM(FDIName,CVHName,MatName)

Airborne mass of material MatName in CVH volume CVHName for FDI location FDIName. The choices allowed for MatName are: ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, ALM – aluminum, ALX – Al_2O_3, UMT – uranium metal, CDM – cadmium, B4C – boron carbide, AIC – Ag-In-Cd, ALL – all components.
(kg)

FDI-SXRAT(FDIName,HSName,Side,MatName)

Oxidation/consumption/generation rate of material MatName on the (LHS or RHS) side Side of HS structure HSName for FDI location FDIName. The choices allowed for MatName are: ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen.
(kg/s)

FDI-SXTOT(FDIName,HSName,Side,MatName)

Integral over time of the

oxidation/consumption/generation rate of the material MatName on the (LHS or RHS`) side of HS structure HSName for FDI location FDIName. The choices allowed for MatName are: ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen.

(kg)

FDI-SRF-POWR(FDIName,HSName,Side)

Heat transfer rate to the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(W)

FDI-SRF-HEAT(FDIName,HSName,Side)

Heat transferred to the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(J)

FDI-TBD-SURF(FDIName,HSName,Side)

Temperature of deposited debris on the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(K)

FDI-SX-ENRGY(FDIName,HSName,Side)

Energy generated by the oxidation of metals on the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(J)

FDI-MASS-SET(FDIName,HSName,Side)

Total mass that has settled out of the atmosphere onto the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(kg)

FDI-ENTH-SET(FDIName,HSName,Side)

Total enthalpy that has settled out of the atmosphere onto the (LHS or RHS) side of HS structure HSName for FDI location FDIName.

(J)

FDI-SRF(FDIName,HSName,Side,MatName)

Mass of material MatName on the (LHS or RHS) side of HS structure HSName for FDI location FDIName. The choices allowed for MatName are: ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, ALM – aluminum, ALX – Al_2O_3, UMT – uranium metal, CDM – cadmium, B4C – boron carbide, AIC – Ag-In-Cd, ALL – all components.

(kg)

6 Example Input

The following input records define a single FDI location. In this example, the FDI location is named FDI15. It interfaces with control volume CV210 and cavity CAV50. The corresponding transfer process input is also shown to help clarify the required input. Material is transferred from the core to FDI through *in* transfer process TP101 and *out* transfer process TP101. Material is transferred from FDI to CORCON through *in* transfer process TP102 and *out* transfer process TP102. Note that any control poison (material 6) that is transferred from COR to TP is eliminated by Transfer Matrix UIN103 before reaching FDI. See the COR, CAV, and TP Users' Guides for more information. The radionuclides are transferred through transfer processes TP601 and TP602. NOTE: Because the FDI_CVH and FDI_DEP records are absent, this input invokes the LPME model only.

SOURCE	IN AND OUT TRANSFER PROCESSES	RECEIVER	MATRIX
COR	$\rightarrow$ TP101 $\xrightarrow{UIN103}$ TP101 $\rightarrow$	FDI	UIN103, 6 x 5
RNCOR	$\rightarrow$ TP601 $\xrightarrow{DEF1}$ TP601 $\rightarrow$	RNFDI	DEF1, NUMCLS x NUMCLS
FDI	$\rightarrow$ TP102 $\xrightarrow{DEF1}$ TP102 $\rightarrow$	CAV	DEF1, 5 x 5
RNFDI	$\rightarrow$ TP602 $\xrightarrow{DEF1}$ TP602 $\rightarrow$	RNCAV	DEF1, NUMCLS x NUMCLS

```
!***** FDI PACKAGE INPUT *****
FDI_INPUT
FDI_ID  FDI15
!       NFDICV NFDCAV NFDTPI NFDTPO
FDI_LOC CV210  CAV  CAV50  TP102  TP101  TP602  TP601
!     ZBOTTM ZTOP
FDI_EL  -5.   0.
...
!**** COR AND CAV PACKAGE INPUT *****
!* NTPCOR
COR_INPUT
!      NTPCOR          RNTPCOR
```

```
COR_TP TP101            TP601
...
CAV_INPUT
CAV_ID CAV50
! NTPOT
CAV_TP   TP102              TP602
...
!***** TP INPUT *****
TP_INPUT
TP_IN   4 ! NAMETPIN *** NAMEPAC DIRECT IP2EDF *** NMSIN NHTRM *
           1  TP101             COR
           2  TP102             FDI
           3  TP601             RNCOR
           4  TP602             RNFDI
TP_OUT  4 ! NAMETPOT ***NPOTOI **** OUTMTX ***** NMSOT**
        1  TP101         TP101       UIN103        5
        2  TP102         TP102       DEF1          5
        3  TP601         TP601       DEF1          16
        4  TP602         TP602       DEF1          16
!***** NAMEMTX ******** NROW NCOL
TP_MTX UIN103             5  6
       ! NROW   Matrix data as a table (NROW,NCOL)
          1      1.0  0.0  0.0  0.0  0.0  0.0
          2      0.0  1.0  0.0  0.0  0.0  0.0
          3      0.0  0.0  1.0  0.0  0.0  0.0
          4      0.0  0.0  0.0  1.0  0.0  0.0
          5      0.0  0.0  0.0  0.0  1.0  0.0
```

The next example shows how to invoke the stand-alone HPME model. Note the use of the EDF input for the debris source and the Material Properties (MP) input to define the composition of the steel.

```
!*****EXTERNAL DATA FILE PACKAGE INPUT *****
!
EDF_INPUT
EDF_ID  EDF1 READ  DCHDATA
EDF_FMT  '(8E9.3)' ! FORMAT OF DATA IN DCHDATA
*
!**** TP INPUT *****
!
TP_INPUT
! 'IN' TRANSFER PROCESS FROM EDF PACKAGE
! READ DCH SOURCE FROM EDF FILE EDF1
!                                 5 MATERIALS, 9 AUX. VARIABLES
TP_IN   1 *NAMETPIN *** NAMEPAC DIRECT IP2EDF *** NMSIN NHTRM *
        1  TP101          EDF    READ   EDF1       5      9
TP_OUT  1 *NAMETPOT ******* NPOTOI ******** OUTMTX ****NMSOT***
        1  TP102            TP101            DEF1        5

!***** FDI PACKAGE INPUT *****
FDI_INPUT
FDI_ID  'CVH300/HS20001'
!       NFDICV NFDCAV NFDTPI NFDTPO
FDI_LOC CV300  NOCAV TPOUT  TP102 ! INVOKE STAND-ALONE
!     ZBOTTM ZTOP
FDI_EL  0.0   5.0 ! ANY TWO REAL VALUES OK
FDI_CVH 1 ! N IDATM ITYP CAV/HS  FATM  TOXV   THT    TST
            1 CV300 LHS  HS20001  1.0  3.E-1 2.E-1  1.E20
FDI_DEP 1 ! N ITYP IDDEP  FDEP   TOXS / NFDITP RNTPINAM
            1 LHS  HS20001 0.0    1.E1
...
!
```

```
!**** MP INPUT *****
!
MP_INPUT
MP_ID 'STAINLESS-STEEL'
MP_SS 1.0 0.0 0.0 0.0 ! 100% FE, no CR, NI, or C
```

The final example shows how the FDI input from the first example can be expanded to activate the normal HPME model. If CVH210 represents the reactor cavity volume with associated CORCON cavity CAV50, the LPME model input in the first example is perfectly reasonable. However, if the reactor vessel were to fail at high pressure, there is a possibility that some of the ejected debris could be blown out of the reactor cavity into the upper containment, in which case HPME model input must be included to treat the situation reasonably. Assume that the nuclear power plant under investigation has been nodalized as depicted in the diagram below. Furthermore, assume that 60% of the ejected debris is deposited directly in the molten pool that forms at the bottom of the reactor cavity (CAV50), 30% splashes into the atmosphere but remains within the reactor cavity (CVH210), 1% is deposited on the walls of the reactor cavity (HS150), and of the 9% that escapes from the reactor cavity, 4% is deposited directly on the walls of the upper containment (HS160), while the remaining 5% is dispersed throughout the upper containment atmosphere. Appropriate values for the time constants for oxidation, heat transfer, and settling in CVH210 and CVH215 can be obtained in two ways—through analysis of the results of an independent DCH calculation of the problem or through simple hand calculations based on reasonable assumptions concerning containment geometry dimensions, particle size, and transfer coefficients (see FDI Reference Manual). Assume values of 0.1 s have been obtained for the oxidation and heat transfer time constants, while a value of 1 s was obtained for the settling time constant. Then the appropriate LPME and HPME input would appear as follows:

```
!
!                 ____________________________
!                 |                          |
!                 |                          |__
!    |        |   |                          |/|
!    | VESSEL |   |          CVH 215         |/| HS
!    | CVH 100|   |                          |/| 160
!    |        |   |                          |/|
!    |        |   |                          |/|
!    |xxxxxxxx|   |                          |-
! ___|xxxxxxxx|___|     ______________________|
! |  \________/        |          |___________|
! |                    |__          HS 155
! |                    |/|
! |       CHV 210      |/|
! |                    |/| HS
! |                    |/| 150
! |                    |/|
! |____________________|-
! \                   /
!  \       CAV 50     /
!   \_________________/
!
!
!**** FDI PACKAGE INPUT *****
FDI_INPUT
FDI_ID  FDI15
```

```
!       NFDICV NFDCAV NFDTPI NFDTPO
FDI_LOC CV210  CAV  CAV50  TP102  TP101  TP602  TP601
!     ZBOTTM ZTOP
FDI_EL  -5.   0.
FDI_CVH 2 ! N IDATM ITYP CAV/HS  FATM  TOXV   THT    TST
            1 CV210 CAV  CAV50   .30   1.E-1 1.E-1 1.E-0
            2 CV215 LHS  HS155   .05   1.E-1 1.E-1 1.E-0
FDI_DEP 4 ! N ITYP IDDEP  FDEP   TOXS / NFDITP RNTPINAM
            1 CAV  CAV50  0.60          TP102   TP602
            2 LHS  HS150  .01    1.E1
            3 LHS  HS155  0.0    1.E1
            4 LHS  HS160  .04    1.E1
```

As might be expected, the *in* transfer process name for the HPME model (NFDITP) is the same as the name associated with the LPME model (NFDTPI) because they transfer material to the same CORCON cavity. Notice that the left-hand sides of heat structures HS150, HS155, and HS160 do not and cannot have values of NFDITP associated with them. It is assumed that HS input will be included to properly orient the left-hand sides of structures HS150, HS155, and HS160 with respect to volumes CV210 and CV215.

In the absence of better information for deriving appropriate time constants for oxidation, heat transfer, and settling for airborne debris, values for these parameters may be estimated as discussed in the following sections.

6.1 Settling Time Constant

The settling time constant in volume *i*, *TST(i)*, may be approximated as

$$TST(i) = L(i)/V(i)$$

where

L(i)	=	settling height in volume *i* (m)
V(i)	=	settling velocity in volume *i* (m/s)

The settling height is limited to the maximum ceiling height in volume *i* but may be less if for some reason the injected debris is not expected to reach that height. The settling velocity may be estimated from the lesser of the value from an appropriate correlation for the terminal velocity of falling bodies and

$$\left[2\, L(i) / g\right]^{0.5} ,$$

where *g* is the acceleration of gravity. Typical values of *L(i)* and *V(i)* would be on the order of 1 to 10 m and 1 to 10 m/s, respectively. Consequently, *TST(i)* is expected to be on the order of 1 s.

6.2 Heat Transfer Time Constant

The heat transfer time constant in volume *i*, *THT(i)*, may be approximated as

$$THT(i) = RHOD \times CPD \times D / (6 \times H)$$

where

RHOD	=	density of debris (kg/m^3)
CPD	=	specific heat capacity of debris (J/kg-K)
D	=	equivalent spherical diameter of debris particles (m)
H	=	debris-to-gas heat transfer coefficient (W/m^2-K)

Typically, *RHOD* is on the order of 10,000 kg/m^3, *CPD* is on the order of 500 J/kg-K, *D* is on the order of 0.001 m and *H* is on the order of 1000 W/m^2-K. Hence, *THT(i)* is expected to be on the order of 0.5 s.

6.3 Oxidation Time Constant

Assuming the oxidation rate is limited primarily by mass transfer in the gas phase and applying the analogy between heat and mass transfer rates in turbulent flow, it is expected that the oxidation time constant in volume *i*, *TOX(i)*, will be approximately equal to *THT(i)*.

NOTE: Because of the uncertainties associated with the values of *TST(i)*, *THT(i)*, *TOX(i)*, and most of the phenomena affecting DCH, it is recommended that users conduct sensitivity studies by varying the assumed values of the HPME model input if the results of a calculation concerning DCH do not conclusively resolve the issues of primary importance.

Appendix 1

MELCOR 2.1	PTFRead	Comments
FDI-FMREL(FDIName) FDIName – the FDI location name.	FDI-FMREL.n n – the FDI location index.	Mass of material released to CORCON from FDI location FDIname (n) for this timestep.
FDI-FMRELT(FDIName) FDIName – the FDI location name.	FDI-FMRELT.n n – the FDI location index.	Integrated mass released to CORCON from FDI location FDIName (n) for entire calculations.
FDI-ETRAN(FDIName) FDIName – the FDI location name.	FDI-ETRAN.n n – the FDI location index.	Energy transferred to water from FDI location FDIname (n) for this timestep.
FDI-ETRANT(FDIName) FDIName – the FDI location name.	FDI-ETRANT.n n – the FDI location index.	Integrated energy transferred to water from FDI location FDIname (n) for entire calculation.
FDI-STGEN(FDIName) FDIName – the FDI location name.	FDI-STGEN.n n – the FDI location index.	Mass of steam generated in FDI location FDIname (n) for this timestep.
FDI-STGENT(FDIName) FDIName – the FDI location name.	FDI-STGENT.n n – the FDI location index.	Integrated steam mass generation for FDI location FDIname (n) for entire calculation.
FDI-OXRAT (FDIName,CVHName,MAT) FDIName – the FDI location name; CVHName – the control volume name; MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen).	FDI-MAT-OXRAT.n.m n – the FDI location index; m – the control volume index; MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen).	Oxidation/consumption/generation rate of material MAT in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-OXTOT (FDIName,CVHName,MAT) FDIName – the FDI location name; CVHName – the control volume name; MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, 'H2' – hydrogen).	FDI-MAT-OXTOT.n.m n – the FDI location index; m – the control volume index; MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, 'H2' – hydrogen).	Integral over time of the material's (MAT) oxidation/ consumption/generation rate in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-ATM-POWR (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-ATM-POWR.n.m n – the FDI location index; m – the control volume index.	Heat transfer rate to atmosphere in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-ATM-HEAT (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-ATM-HEAT.n.m n – the FDI location index; m – the control volume index.	Heat transferred to atmosphere in CVH volume CVHName for FDI location FDIName.

MELCOR 2.1	PTFRead	Comments
FDI-DEBRIS-T (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-DEBRIS-T.n.m n – the FDI location index; m – the control volume index.	Airborne debris temperature in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-OX-ENRGY (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-OX-ENRGY.n.m n – the FDI location index; m – the control volume index.	Energy generated by the oxidation of zircaloy and steel in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-MASS-ADD (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-MASS-ADD.n.m n – the FDI location index; m – the control volume index.	Mass transferred from TP package to CVH volume CVHName (m) for FDI location FDIName (n).
FDI-ENTH-ADD (FDIName,CVHName) FDIName – the FDI location name; CVHName – the control volume name.	FDI-ENTH-ADD.n.m n – the FDI location index; m – the control volume index.	Enthalpy transferred from TP package to CVH volume CVHName (m) for FDI location FDIName (n).
FDI-ATM (FDIName,CVHName,MAT) FDIName – the FDI location name; CVHName – the control volume name; MAT – the material name (ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, 'ALM' – aluminum, ALX – Al_2O_3, UMT – uranium metal, CDM – cadmium, B4C – boron carbide, AIC – Ag-In-Cd, ALL – all components).	FDI-ATM-MAT.n.m n – the FDI location index; m – the control volume index; MAT – the material name (ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, ALM – aluminum, ALX – Al_2O_3, 'UMT' – uranium metal, CDM – cadmium, B4C – boron carbide, AIC – Ag-In-Cd, ALL – all components).	Airborne mass of material MAT in CVH volume CVHName (m) for FDI location FDIName (n).
FDI-SXRAT (FDIName,HSName,Side,MAT) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS); MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen).	FDI-MAT-SXRAT.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS); MAT – the material name (ZR– zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST– steam, H2 – hydrogen).	Oxidation/consumption/generation rate of material MAT on the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).

MELCOR 2.1	PTFRead	Comments
FDI-SXTOT (FDIName,HSName,Side,MAT) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS); MAT – the material name (ZR– zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen).	FDI-MAT-SXTOT.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS); MAT – the material name (ZR – zircaloy, AL – aluminum, SS – steel, O2 – oxygen, ST – steam, H2 – hydrogen).	Integral over time of the oxidation/consumption/generation rate of the material MAT on the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-SRF-POWR (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or 'RHS).	FDI-SRF-POWR.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS).	Heat transfer rate to the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-SRF-HEAT (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS).	FDI-SRF-HEAT.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS).	Heat transferred to the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-TBD-SURF (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS).	FDI-TBD-SURF.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS').	Temperature of deposited debris on the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-SX-ENRGY (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS).	FDI-SX-ENRGY.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS).	Energy generated by the oxidation of metals on the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).

MELCOR 2.1	PTFRead	Comments
FDI-MASS-SET (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS).	FDI-MASS-SET.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS).	Total mass that has settled out of the atmosphere onto the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-ENTH-SET (FDIName,HSName,Side) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS).	FDI-ENTH-SET.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS).	Total enthalpy that has settled out of the atmosphere onto the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).
FDI-SRF (FDIName,HSName,Side,MAT) FDIName – the FDI location name; HSName – the heat structure name; Side – the heat structure side (LHS or RHS); MAT – the material name (ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, ALM – aluminum, ALX – Al_2O_3, 'UMT' – uranium metal, CDM – cadmium, B4C – boron carbide, AIC – Ag-In-Cd, ALL – all components).	FDI-SRF-MAT.s.n.k n – the FDI location index; k – the heat structure index; s – the heat structure side (LHS or RHS); MAT – the material name (ZRM – zircaloy, ZRX – ZrO_2, UO2 – UO_2, SSM – steel, SSX – steel oxide, ALM – aluminum, ALX – Al_2O_3, UMT – uranium metal, CDM – cadmium, B4C – boron carbide, 'AIC – Ag-In-Cd, ALL – all components).	Mass of material MAT on the side Side (s) of HS structure HSName (k) for FDI location FDIName (n).

Flow Path (FL) Package
Users' Guide

The MELCOR Control Volume Hydrodynamics (CVH) and Flow Path (FL) packages model the thermal-hydraulic behavior of liquid water, water vapor, and gases in MELCOR. Modeling is based on a control volume/flow path formulation and is described in detail in the Thermal Hydraulic (CVH and FL) Packages Reference Manual.

This users' guide describes input to the FL package. The package is concerned with the geometry and characteristics of connections between control volumes through which the control volume contents may be transported. FL input describes these connections and defines the network through which hydrodynamic materials (water, water vapor, and noncondensible gases residing in control volumes) may flow. No material resides in flow paths. When a flow of atmospheric materials (water vapor and noncondensible gases) enters a control volume below the surface of a water pool in that volume, the resulting mass and heat transfer will be calculated if requested by user input to the FL package.

The phenomena of thermal equilibration and condensation of evaporation are treated as occurring within the flow path, using models from the SPARC90 code.

In addition to geometry, FL input includes all definition of flow resistance, including any frictional losses associated with the walls of control volumes and blockages calculated by the Core (COR) package. Special models including externally controlled flow areas (valves), forced flows, and momentum sources (pumps) are also defined by FL input in conjunction with functions defined by the Tabular Function (TF) and Control Function (CF) packages.

The geometry and content of control volumes are defined by input to the CVH package, as described in the CVH Package Users' Guide.

Contents

CONTENTS **2**

LIST OF FIGURES **5**

1 FLOW PATH MODELS **6**

1.1 Flow Path Definition 6
1.1.1 Area and Length 7
1.1.2 Horizontal and Vertical Paths, Junction Openings 8
1.1.3 Gravitational Head Terms 9
1.2 Frictional Losses 10
1.2.1 Form Loss Coefficients 10
1.2.2 Flow Path Segments and Wall Friction 11
1.3 Two-Phase Flow 12
1.3.1 Void Fraction 12
1.3.2 Momentum Exchange Length 13
1.4 Choked Flow 14
1.5 Bubble (SPARC) Physics 14
1.6 Other Models 15
1.6.1 Valves 15
1.6.2 Pumps and Fans 17
1.6.3 Mechanical Model 18
1.6.4 Heat Exchanger Model 18
1.6.5 Time Dependent Flow Paths 18
1.6.6 Pool-First and Other Flow Path Options 19
1.6.7 Momentum Flux Terms 19
1.6.8 Flow Blockage Controlled by Other Packages 19
1.6.9 Flashing of Superheated Pool Flows 20
1.6.10 New Stratified Counter Current Flow Model 20
1.7 Flow Paths to and from Core Volumes 20
1.8 Modeling Breaks and Failures 21
1.9 MACCS Interface 22
1.9.1 MACCS Releases by Ring (or SFP Batch) 22

2 USER INPUT 22
2.1 MELGEN User Input 23
2.1.1 FL Records Associated With a Particular FL_ID record 23
FL_ID – Flow Path Name and Sequence Number 23
FL_DELETE – Ignore all records associated with FL_ID 24
FL_FT – Control Volume Data that the Flow Path Connects 24
FL_GEO – Flow Path Geometry 25
FL_JSW – Flow Path Junction Switches 26
FL_USL – User Specified Loss Coefficients 28
FL_VEL – Initial Atmosphere and Pool Velocities 29
FL_LME – Length for Pool/Atmosphere Momentum Exchange 29
FL_EDF – Flow Path Connection to EDF 30
FL_JLF – Junction Limits for "*From*" Volume 30
FL_JLT – Junction Limits for "*To*" volume 31
FL_BLK – Data for Blockage of Flow by Another Package 31
FL_SEG – Piping Segment Parameters 33
2.1.2 FL Records Not Associated With a Particular FL_ID record 35
FL_FLSH – Flow Path Flashing for Forward (Positive) and Reverse (Negative) Flows 35
FL_VLV – Valve Input 37
FL_PMP – Pump Input Data 39
FL_VTM – Time Dependent Flow Path 41
FL_MACCS – MACCS Release Paths Definition 42
FL_MACCS_SFP1 - MACCS Spent Fuel Pool Definition 1. 45
FL_MACCS_SFP2 - MACCS Spent Fuel Pool Definition 2 46
FL_MACCS_SFP3 - MACCS Spent Fuel Pool Definition 3 46
FL_MFX – Momentum Flux Input Data 47
FL_MCH – Mechanical model 48
FL_IHX – Heat Exchanger Model 49
FL_CCF – Counter-Current Stratified Flow Model 50
2.2 MELCOR User Input 52

3 SENSITIVITY COEFFICIENTS 53

4 PLOT VARIABLES 53

5 CONTROL FUNCTION ARGUMENTS 56

6 EXAMPLE INPUT 60

7 FL PACKAGE OUTPUT 63

8 DIAGNOSTICS AND ERROR MESSAGES 64

APPENDIX 1 65

List of Figures

Figure 1.1 Flow Path Definition 6
Figure 1.2 Junction Geometry in Tank-and-Tube Limit. 8
Figure 1.3 Junction Geometry in Separating-Surface Limit 9
Figure 1.4 Flow Path Segments 11
Figure 1.5 Vertically Stacked Control Volumes with Stratification 13
Figure 1.6 Flow Paths for "*To*" and "*From*" Core Volumes 21

1 Flow Path Models

The Flow Path (FL) Package Users' Guide presents only an overview of the modeling in the MELCOR FL package. More detailed descriptions, with appropriate references, may be found in the Thermal Hydraulic (CVH and FL) Packages Reference Manual.

1.1 Flow Path Definition

Each flow path connects two control volumes. Each connection is referred to as a junction; the two junctions associated with a flow path may be at different elevations. One volume is referred to as the "*From*" control volume and the other as the "*To*" control volume, thus defining the direction of positive flow. See Figure 1.1.

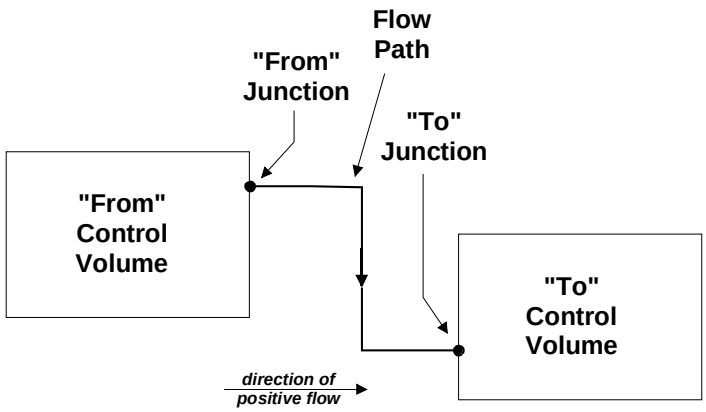


Figure 1.1 Flow Path Definition

Because there is no residence time for material flowing through a flow path, there is no mass or energy associated with a flow path; all mass and energy reside in the control volumes. There is no heat or mass transfer between the pool and atmosphere materials flowing through the flow paths, nor is there heat transfer to or from structures since heat structures are not allowed to interact with flow paths.

A flow path may represent a pipe-like connection in a tank-and-tube model or the open area of a separating surface (cell boundary) in a finite-difference-like model. The former case represents the limit of a control volume when residence effects are not important. In the interest of computational speed, MELCOR nodalizations are typically made relatively coarse, with only a modest number of control volumes. Therefore, it is common to reduce volumes of intermediate size with potentially high flow, such as relief valve discharge lines, pressurizer spray lines in a pressurized water reactor (PWR), or jet pumps in a boiling water reactor (BWR), to simple flow paths in a MELCOR nodalization. The associated volume is typically included in one of the connected control volumes.

The same input parameters, differently chosen, are used to represent the two limiting types of flow paths (pipe-like and cell boundary) and a variety of intermediate cases.

The primary difference is in the definition of the junctions of control volumes. The junction elevations represent the elevations of the central points of the connections to the respective control volumes; the junction opening heights represent the range of elevations over which material can be drawn out of the corresponding volume into the flow path.

For a pipe-like flow path, the junction elevations are typically unequal and the opening heights should be characteristic of the dimensions of the pipe. For a cell-boundary flow path, the junction elevations should be chosen to be equal, defined by the elevation of the mid-point of the cell boundary, and each opening height should be characteristic of the dimensions of the associated control volume.

1.1.1 Area and Length

Among the fundamental properties of a flow path are its area and its length. In many cases, a flow path represents a geometry with varying flow area. In most cases, the area input as *FLARA* on the FL_GEO record in FL input should be chosen as the minimum area along the path.

A flow path need not be fully open. The fraction *F (0.0 ≤ F ≤1.0)* that is initially open may be defined by the variable FLOPO on the FL_GEO record. *F* may also be modified as a function of time by a valve model in the flow path, as described in Section 1.6.1.

The CVH package calculates a velocity for each phase, pool and atmosphere, in each flow path. Only the open area, $F \cdot FLARA$, is used in the models: the volume flow $J \equiv F \cdot FLARA \cdot v$ is used in the advection of materials and in definition of wall friction (Section 1.2.2), and the mass flux based on the open area is the one compared with the critical mass flux in the choking model (Section 1.4). The individual variables *F* and *FLARA* may clearly be chosen in several ways. It is common, but not universal, in constructing FL package input to choose *FLARA* as the maximum area that will <u>ever</u> be open in the flow path. *FLOPO* will then be 1.0 for all flow paths not containing valves, and a fully open valve will correspond to an open fraction of 1.0.

The length specified as *FLLEN* on the FL_GEO record in the FL input will be used as an inertial length. The inertia of a flow path is a measure of the average mass per unit area along its length. If the area is not constant, a rigorous approach is to choose the length input as *FLLEN* in conjunction with the flow path area, *FLARA*, to match this average:

$$\frac{FLLEN}{FLARA} = \int_{"fm"}^{"to"} \frac{dx}{A(x)} \tag{1.1}$$

Here *x* measures distance along a path from the center of the "*From*"' volume to the center of the "*To*" volume, and *A(x)* is the flow area at *x*. In practice, this is rarely necessary.

FLLEN may also be used as the default length over which the force between pool and atmosphere acts. See Section 1.3.2 for details. It is not used in friction calculations; segment lengths are used instead, as discussed in Section 1.2.2.

1.1.2 Horizontal and Vertical Paths, Junction Openings

Flow paths are designated as horizontal or vertical, depending on the dominant direction of flow in the path, by user input on record FL_JSW; the default is vertical. The designation affects the definition of junction geometry and the default definition of the momentum exchange length in two-phase flow (Section 1.3.2).

The nominal elevation of each junction, defined by *ZFM* or *ZTO* on input record FL_FT, represents the midpoint of the connection and must lie within the range of elevations contained in the volume. The junction opening height defines the range of elevations, *H*, over which material may be drawn out of a control volume through a flow path. By default, as discussed below, this range is calculated from a total junction height based on the flow path area and orientation; this total height may be over-ridden by user input of *FLHGTF* or *FLHGTT* on record FL_GEO. In either case, if the top and/or bottom of the junction opening, at elevation $Z \pm H/2$, lies outside of the control volume, the junction opening for a horizontal flow path will be truncated; for a vertical flow path, it will first be translated to preserve as much as possible of the original height *H*.

The default values of junction opening heights (*H*), as shown in Figure 1.2, are the diameter of a circle of area *FLARA* for a horizontal path and the radius of such a circle for a vertical path. These values are appropriate for a tank-and-tube model, as shown in Figure 1.2. For a horizontal flow path, the default represents the range of altitudes seen by a circular pipe; for a vertical flow path, the default represents a rough estimate of the extent to which the pool surface may be drawn down by flow through a vertical drain.

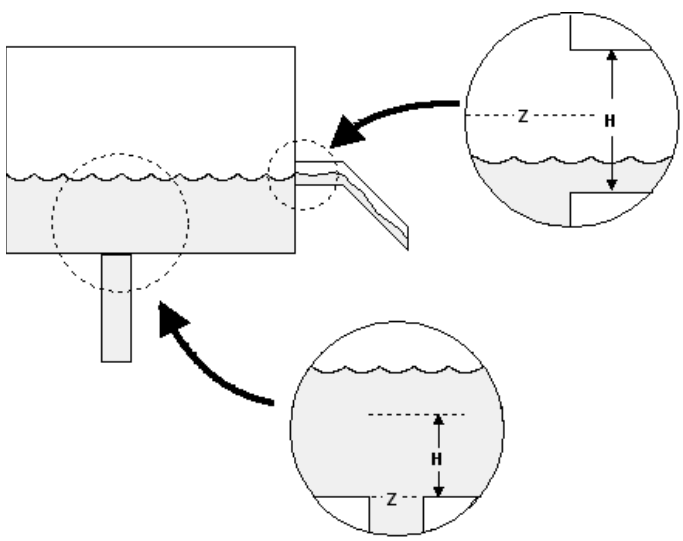


Figure 1.2 Junction Geometry in Tank-and-Tube Limit.

As an alternative, the user may elect to specify the elevations of the top and bottom of the junction opening to either or both control volumes directly on FL_JLF and/or FL_JLT records. (These records may also be used to modify the junction opening on restart.) If these records are used, no adjustments will be made. The bottom of the opening must lie between the bottom of the control volume and the nominal junction elevation

(inclusive), while the top must lie between the nominal junction elevation and the top of the control volume (inclusive).

The elevations of the top and bottom of each flow path, whether directly input or calculated, are included in the "FLOW PATH TIME INDEPENDENT EDIT" generated by MELGEN and reproduced at the beginning of each MELCOR run.

A flow path may also represent an open surface separating two control volumes. For a horizontal flow path, corresponding to a vertical surface, the opening height should be taken to include the entire open area. The elevation of the center of the opening should be used for both junction elevations, as shown in Figure 1.3. It is generally necessary to override the default opening height to ensure that the entire open area—and only that area—lies within the junction opening. For vertical flow through a horizontal separating surface, the junction elevations should again be chosen to be equal. The opening height has no rigorous interpretation; it serves only to define the range of elevations from which material may be drawn, and some significant fraction of the volume height is appropriate. Because the flow equations include both gravitational (buoyancy) and momentum exchange (entrainment) forces, slip between the phases will tend to create and preserve stratification of the pool and atmosphere stratification unless velocities are great enough to cause entrainment. Thus, extremely small vertical opening heights are not necessary to preserve the stratification of pool and atmosphere in cases where it would be expected to occur. See Section 1.3.1 for further discussion in the context of the two-phase flow.

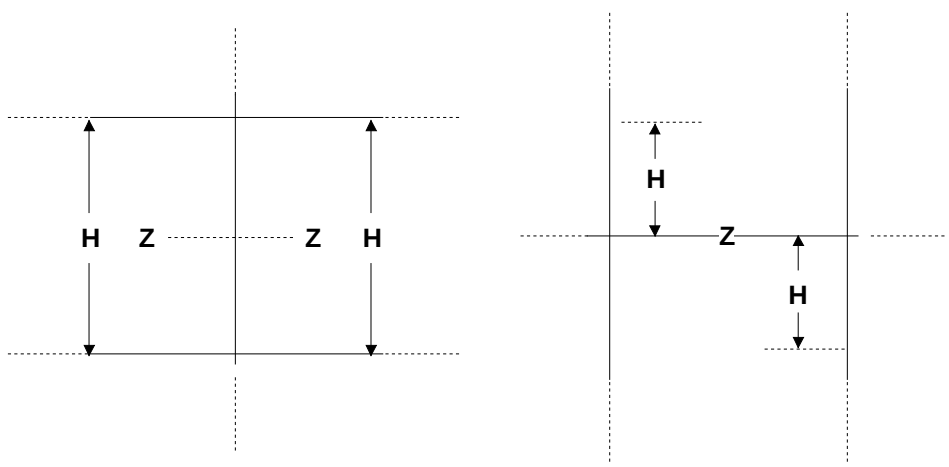


Figure 1.3 Junction Geometry in Separating-Surface Limit

1.1.3 Gravitational Head Terms

The equation for each flow includes a gravitational head term, calculated as the sum of three contributions:

(1) the head difference between the reference point for pressure in the "*From*" volume and the junction in that volume;

(2) the head difference associated with the change of elevation within the flow path; and

(3) the head difference from the junction in the "*To*" volume and the reference point in that volume.

The pressure in a MELCOR control volume is defined at the pool/atmosphere interface, taken at the bottom of the volume if there is no pool, and at the top of the volume if there is no atmosphere.

For a pipe-like flow path with junctions at different elevations, within-path contribution is particularly significant. There is a basic limitation in its evaluation: the flow path has no contents and, thus, no density or temperature of its own. A control volume density (or an average) must, therefore, be used for the within-path contribution to the head. This is adequate for those purposes where the net head results from the difference between pool and atmosphere density. In particular, it properly accounts for the additional head which must be overcome to depress a liquid surface and clear sparger vents or to raise a liquid surface to initiate overflow of a standpipe. It is not adequate for calculation of natural convection where the net head results from the change in density of one phase as a function of temperature.

Therefore, if a natural convection loop involving a number of control volumes is expected to be an important part of a MELCOR calculation, flow paths connecting points with different elevations should not be used to represent connections through which the circulating flow will pass.

1.2 Frictional Losses

All dissipative pressure drops between volumes are assumed to take place within the flow paths connecting them. Contributions from both form loss and wall friction are included. An additional frictional term calculated by the Core (COR) package may be included where appropriate to model the effect of blockage by core debris, as described in Section 1.6.8.

1.2.1 Form Loss Coefficients

The form loss calculation is based on user-input loss coefficients (K in Equation 1.2), which may be different for forward and reverse flow. These coefficients are input on the FL_USL record and are applied directly to the velocities calculated by the CVH and FL packages. The resulting pressure differential (in the direction of positive flow) is

$$\Delta P_{\varphi} = -\frac{1}{2} K \rho_{\varphi} \left| v_{\varphi} \right| v_{\varphi} \tag{1.2}$$

for phase ϕ, where $\phi = P$ or A to denote pool or atmosphere, respectively.

For a complicated pipe network, the individual contributions from multiple changes of area must be combined into a single loss coefficient to be used in Equation (1.2).

1.2.2 Flow Path Segments and Wall Friction

A wall friction contribution is added to the form loss term. In evaluating this contribution, the flow path is treated as consisting of one or more segments in series; this allows an accurate representation of wall friction in complicated pipe-like geometries.

The flow path velocity and open area are used with the segment area to calculate a distinct velocity within each segment, based on the assumption of incompressible flow. The segment velocity is then given by

$$v_s = FAv/A_s \tag{1.3}$$

(F is again the fraction of the nominal area that is open), so that the segment velocity depends only on the volumetric flow through the flow path, $J \equiv FAv$. The frictional pressure differential is evaluated as

$$\Delta P_\varphi = -\sum_s \left(2f_s L_s / D_s\right)\rho_\varphi \left|v_{\varphi,s}\right| v_{\varphi,s} \tag{1.4}$$

where L_s and D_s are the length and hydraulic diameter of the segment. The Fanning friction coefficient, f_s, is calculated according to the method of Beattie and Whalley, using a mixture Reynolds number based on the segment velocity.

The segments should represent all important aspects of the flow path, from volume center to volume center. If wall losses within a volume are important, the geometry of the appropriate parts of the volume should, therefore, be included as segments in the flow path because all frictional losses are assumed to occur within the flow paths.

Figure 1.4 illustrates the use of flow path segments. Volumes "*From*" and "*To*" are connected by a pipe manifold, which will be represented as a flow path. For the situation depicted, input for segment 1 should be based on the flow area, hydraulic diameter, and half length of the "*From*" volume and input for segment 4 on the same properties for the "*To*" volume. Input for segment 2 should use the length and hydraulic diameter of a single tube and the total area, which is three times the area of a single tube. Appropriate tube properties should be used for segment 3. Note that no material resides in the flow path; if the volume of the manifold is significant, it should be accounted for by increasing the volumes of "*From*" and/or "*To*".

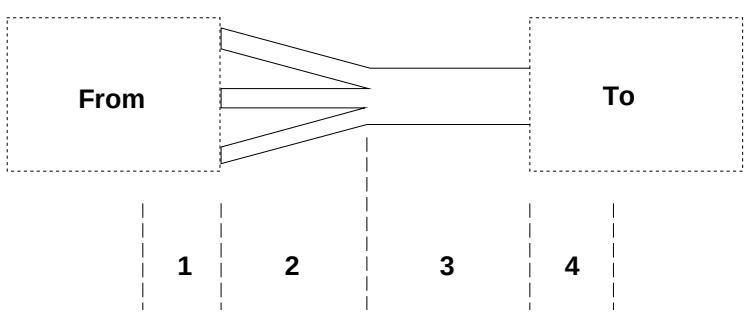


Figure 1.4 Flow Path Segments

The segmentation of a flow path affects frictional losses only; the segment lengths and areas are not used by any other models.

1.3 Two-Phase Flow

The cross-sectional area of a flow path may be shared by flows of pool and atmosphere materials. If both are present, their velocities will not, in general, be equal, and countercurrent flow is possible. The open flow area is partitioned between them in accordance with a calculated void fraction, α, which should, strictly speaking, be referred to as an "atmosphere fraction".

Separate equations are solved for the two velocities. A momentum exchange term is included to model the forces acting between the two flows. A balance among this term, gravitational terms, and friction terms is responsible for the prediction of unequal velocities in quasi-steady flow.

1.3.1 Void Fraction

The void fraction in a MELCOR flow path is defined as the fraction of the open area of the path that is occupied by atmosphere flow. For concurrent flow of pool and atmosphere, it is taken as the fraction of the upstream junction opening that is occupied by atmosphere. For countercurrent flow, a weighted average of the area fractions at the two junctions is used.

This effectively prevents flow of pool out of a control volume unless the pool surface is above the bottom of the junction opening in that volume. Similarly, atmosphere is prevented from flowing unless the pool surface is below the top of the opening. Junction opening heights must be specified to ensure that such trapping of pool or atmosphere will occur only where it is physically appropriate.

MELCOR calculations sometimes show a small water pool in the upper of a pair of vertically stacked control volumes, with an atmosphere in the volume below. Users sometimes specify extremely small opening heights, H, for the vertical flow path—that defines a separating surface between them—in an attempt to produce near-perfect separation of the phases (see Figure 1.5). We have not found this approach to be particularly effective and have seen evidence that it contributes to numerical problems and slow running of MELCOR. Use of somewhat larger opening heights frequently leads to a more orderly draining of the pool into the lower control volume.

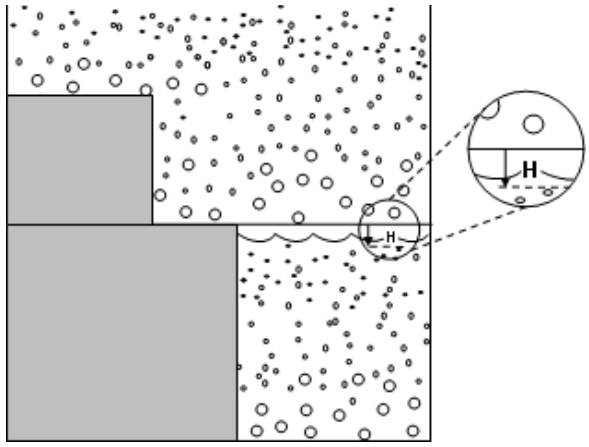


Figure 1.5 Vertically Stacked Control Volumes with Stratification

1.3.2 Momentum Exchange Length

As noted above, the flow equations include both gravitational separation (buoyancy) and momentum exchange (entrainment) forces. A simple model is used for the momentum exchange; it was chosen to reproduce the Wallis flooding curve in the appropriate limit and is described in the CVH/FL Reference Manual. The extent of coupling or separation between the flows is strongly influenced by the relative magnitudes of these forces, which are proportional to the distances over which they act. For horizontal flow paths, the default length over which momentum exchange occurs is taken as the inertial length of the flow path. For vertical flow paths, the default length is taken as the difference in elevation between the lowest point (including the junction opening) and the highest point in the flow path. The different treatment of vertical paths is intended to make the two forces act over the same distance for vertical paths which represent the boundaries of vertically stacked control volumes.

These defaults are appropriate for simple vertical pipes and not unreasonable for horizontal pipes. However, they can lead to significant overprediction of coupling between the phases—and underprediction of countercurrent flow—in a number of cases where a flow path includes a number of separated channels. Examples include the entrance or exit paths for a reactor core and perhaps steam generator tube bundles, where the two phases may flow through distinct, non-interacting portions of the actual geometry.

The default length may be replaced by a user-input length *XL2PF* on the optional FL_LME input record. Increasing the length will tend to tighten coupling between the flows, reducing both the relative velocities and the limits on countercurrent flow; decreasing the length will have the opposite effect. Reducing the length for core flow paths will tend to reduce the levitation of water sometimes predicted by MELCOR in the upper plenum and core regions after lower regions have voided. A momentum exchange length of zero can be used to model a "perfect" separator in steam generator secondaries.

1.4 Choked Flow

The critical mass flux for pool flow in MELCOR is based on the RETRAN model; it uses the Moody model for saturated water and the Henry-Fauske model for subcooled water, with a small interpolation region between them. A further interpolation to simple orifice flow is imposed at large subcooling. The critical mass flux for atmosphere flow is based on sonic flow at the minimum section. An interpolation scheme motivated by the Moody model is used for mixed flows. The mass flux is evaluated based on the open area of the flow path (the fraction open times the nominal area).

The possibility of choked flow is automatically modeled in all flow paths. No user input is required, but the default forward and reverse discharge coefficients of 1.0 may be modified, if desired. The choking model restricts the mass flux through a flow path (based on the open area) to be no greater than the critical value multiplied by the discharge coefficient corresponding to the current flow direction. A discharge coefficient less than 1.0 may be used to account for vena contracta effects. The possibility of choking in a given flow path may be effectively eliminated by specifying a value substantially greater than 1.0 for the associated discharge coefficient.

If choking is detected, pool and atmosphere velocities are set equal. See the Thermal Hydraulics (CVH and FL) Packages Reference Manual for more details.

1.5 Bubble (SPARC) Physics

If a flow of atmospheric material enters a control volume below the surface of the pool in that volume, the material must pass through the pool to reach its final destination. By default, any interactions during this passage are ignored and the atmospheric material is simply added to the control volume atmosphere.

The user may specify that interactions between the pool and the rising gases be modeled by setting one or both of the flags IBUBF and/or IBUBBT on the FL_JSW input record. Selection of the interaction model may be made independently for the two flow directions. If the model is active, the RadioNuclide (RN) package will perform a pool scrubbing calculation using the SPARC90 model to evaluate the removal of fission products from rising gases for those flow paths and flow directions—and only those paths and directions—for which the model is activated on the FL_JSW record. No additional input (other than activation of the RN package) is required, and RN cannot be made to perform a pool scrubbing calculation for a flow path where SPARC modeling was not selected in the FL package input.

The condition of the transmitted gases and the receiving pool is modified by mass and heat transfer processes. The interaction process is viewed as involving the breakup of the injected gas stream into a swarm of bubbles, thermal equilibration of the gases with the pool, and saturation of the bubbles with water vapor at local conditions. This process is treated parametrically. If there are no noncondensible gases in the flow, the bubbles may be completely condensed. See Section 6.1 of the CVH/FL Reference Manual for further details of the model. The efficiency of the process is controlled by sensitivity coefficients in the array C4405.

The interaction is treated as occurring within the flow path. The bubbles are not considered to be resident in the pool for the purposes of calculating the pool composition and do not contribute to pool swelling.

1.6 Other Models

Several additional models are available for use in flow paths. These can be used to (1) modify the fraction of the flow path area that is open, (2) define momentum sources in the path, (3) specify the velocity in the path, (4) modify the treatment of two-phase flow, (5) include the effects of momentum flux, or (6) model blockage of the path by materials such as core debris, or (7) model the stratified countercurrent flow of gases.

1.6.1 Valves

The user may include a valve in a flow path to control the fraction of the area of the path that is open. The open area of the flow path is defined as the fully open area (*FLARA*) multiplied by the fraction open. Valves do not modify the areas of flow path segments and, therefore, do not affect the dependence of wall friction on volumetric flow.

For the specific case of an ideal check valve that allows flow in one direction only, an alternate approach is available in MELCOR, where the definition of flow path types has been extended to include one-way flow paths. This approach avoids the limitations of the numerically explicit nature of the general valve model, including the numerical difficulties that may arise when a user specifies a very large reverse form loss coefficient in an attempt to restrict any calculated reverse flow through a check valve that was open at the start of an advancement. If, however, it is essential to model the differential pressure necessary to open a check valve, valve inertia, or hysteresis, then the general approach described below must be used.

Control functions are used to define the fraction open; values returned will be bounded internally to lie between 0.0 and 1.0 before they are used. When the fraction open is 0.0, the flow path is closed. When the fraction open is 1.0, the flow path is fully open, and the open area is equal to user input flow path area *FLARA*. Two approaches are permitted. In simple cases, the user can specify a single control function that defines the fraction of the area that is open. In particular, a relief valve is typically modeled in MELCOR by defining the open fraction as a hysteresis function of the pressure differential between volumes connected by the flow path. See Section 7.3 of the Control Function (CF) Package Users' Guide for examples of control function input.

In more complicated situations, a trip control function can be specified that is used to switch among three definitions of open fraction as follows:

(1) If the trip control function is 0.0, the trip is "*off*" and the area fraction is unchanged from its last value;

(2) If the trip control function is positive, the trip is "*on-forward*" and the area fraction is that defined by an "*on-forward*" control function; and

(3) If the trip control function is negative, the trip is "*on-reverse*" and the area fraction is that defined by an "*on-reverse*" control function.

See the CF Package Users' Guide for more information on trip control functions.

Losses in a valve can be calculated using the traditional loss coefficient method for an area transition where the form loss calculation is based on a dimensionless user-input loss coefficients (K), which may be different for forward and reverse flow and is used to model pressure losses for various pipe fittings. These coefficients are input on the FL_USL record and are applied directly to the velocities calculated by the CVH and FL packages and are assumed to be constant over time. The resulting pressure differential (in the direction of positive flow) is incorporated into the Darcy-Weisbach equation to include wall friction. The dimensionless number (K) is used to characterize the fitting(s) independent of the properties of the pipe:

$$\Delta P = \left(\frac{fL}{D} + \sum K\right)\frac{\rho v^2}{2} \tag{1.5}$$

Note that the sum of the resistance coefficients (ΣK) is not multiplied by the Moody friction factor f.

An alternative method for prescribing flow losses in a valve is through the valve flow coefficient (C_v) method. The coefficient is dynamic and can vary with the valve opening. By definition, a valve has a C_v of 1 when a pressure of 1 psi causes a flow of 1 US gallon per minute of water at 60°F, i.e. specific gravity (SG) is equal to 1 through the valve. Since the pressure drop through a valve is proportional to the square of the flow rate the relationship between C_v, flow rate and pressure drop can be expressed as:

$$C_v = Q * \sqrt{\frac{SG}{\Delta P}} \tag{1.6}$$

Where,

Q = Flow in gallons per minute (gpm)

C_v = Valve flow coefficient

ΔP = Difference in pressure between inlet and outlet (psi)

SG = specific gravity of liquid relative to water at 60 F.

From this equation it is seen that it is possible to convert between C_V and loss coefficient (K) enabling us to define a Cv for any fitting. Similarities between equations 1 and 2 can then be used to derive the relationship between K and C_V:

$$C_v = \frac{D^2}{\sqrt{K}} * \left[\frac{\pi}{4}\sqrt{\frac{2}{\rho_0}}\right] \tag{1.7}$$

where,

ρ_0=density of water at 60 F.

D = the pipe diameter

1.6.2 Pumps and Fans

Pumps can be included in flow paths. They are modeled as introducing a "pressure boost", which is ordinarily a function of the volumetric flow through the path. In defining a pump, the flow should be thought of as the independent variable and the pressure head delivered by the pump as the dependent variable; the actual flow on any timestep is calculated from the balance of this head (as a function of flow) against static, frictional, and acceleration pressure differentials in the rest of the flow circuit.

There are two types of pumps available in MELCOR. The first, referred to as "FANA", was originally intended to model a fan that impels the atmosphere through a flow path. It can also be used to represent a constant speed coolant pump, although in a very simple form. The second type of pump, called "QUICK-CF", simply uses a control function to define the pressure head, allowing the user complete freedom but also giving the user complete responsibility for all details.

Both models are numerically explicit; that is, the pressure head is based on conditions at the start of the MELCOR system timestep and remains unchanged throughout the step—even if the CVH package is subcyling.

The FANA model assumes that the fan can be adequately represented by a parabolic pressure head versus volumetric flow rate equation. The velocity of the pool is set equal to that of the atmosphere when the pump is on, so that the pump will move the pool through a flow path if the void fraction in the flow path is less than 1.0. The major limitation in the model is the absence of any direct dependence of the pressure head on the density of the flowing material. It should be noted that the intention of FANA pump was designed for atmosphere flow only, but has been adapted for moving both liquid and gas as well. Declaring forward or backward flow only is allowed. The users should be cautioned to use this pump method. Additionally, it assumes that the velocity of the pool and atmosphere is identical. Therefore, KFLGFL=6 for the vertical flow and KFLGFL=7 for the horizontal flow are the acceptable flow paths. Any other flow types input will result in termination of the input processing and no restart will be generated.

The user must define the maximum pressure head, *DPMAX*, and the volumetric flow, *VPZERO*, at which the head is zero. The user may optionally define a non-zero volumetric flow, *VPMAX*, corresponding to the point of maximum head generation; otherwise, maximum head is assumed to occur at zero flow. If *DPMAX* and *VPZERO* are both positive, the fan produces a pressure boost in the direction of positive flow, corresponding to a fan blowing from the "*From*" volume to the "*To*" volume. The pressure head is then given by,

$$\frac{DP}{DPMAX} = \begin{cases} 1.0 & VDOT \le VPMAX \\ \sqrt{\dfrac{VPZERO - VDOT}{VPZERO - VPMAX}} & VPMAX \le VDOT \le VPZERO \\ 0.0 & VPZERO \le VDOT \end{cases} \tag{1.8}$$

where *DP* and *VDOT* are the pressure head and volumetric flow, respectively. A fan that is oriented in the opposite direction, producing its head in the direction of negative flow, can be modeled by simply defining *DPMAX* and *VPZERO* to be negative, and reversing the sign of *VPMAX*, if it is input. In this case, the effect of the pump is maximum for $VDOT \ge VPMAX$ and is zero for $VDOT \le VPZERO < 0$.

There is no provision to control the speed of the fan, but it can be turned on and off, if desired, by defining either a control function or a tabular function as a function of time that will be zero when the fan is to be off and non-zero when it is to be on. By default, the fan is always on; when it is, *DP* is simply set to zero.

Note that this model is not a reasonable representation of a real pump far outside the range of flows given by $VPMAX \le VDOT \le VPZERO$.

The QUICK-CF pump model relies on a user-defined control function to evaluate the pressure boost (in Pa) for the pump. All control logics must be included in the control function. Section 7.7 of the CF Package Users' Guide provides an example of input for a homologous pump similar to TRAC and RELAP.

1.6.3 Mechanical Model

The mechanical model simulates a turbine or compressor in a flow path by adding a constant pressure boost to the flow path. The isentropic efficiency and ideal gas properties are used to determine the corresponding temperature increase (or drop).

1.6.4 Heat Exchanger Model

MELCOR contains an integral model for heat exchangers. In previous versions, the user was required to construct a heat exchanger using control volumes, flow paths, and heat structures. A rather detailed nodalization was often required to capture the temperature profiles within heat exchanger tubes and/or shells. The necessarily small size of the volumes often forced the code to take very short time steps.

In contrast, the integral model simply connects two flow paths, and implicitly accounts for the temperature profiles within the heat exchanger primary and secondary. Heat is transferred *within* the flow paths, using quasi-steady relationships familiar from the "Number of Transfer Units" (NTU) formulation of heat exchanger performance. The heat removal or addition is effectively imposed on the actual downstream volume in each flow path. See Section 6.10 of the CVH/FL Reference Manual for further details.

1.6.5 Time Dependent Flow Paths

The velocities in a flow path can be specified by the user through control functions or tabular functions. The specified velocity is used for both the atmosphere and the pool. The void fraction is calculated using the standard model.

1.6.6 Pool-First and Other Flow Path Options

Several modifications may be made to the treatment of two-phase flow described in Section 1.3. These are activated by choice of the *KFLGFL* flag on input record FL_JSW, the same flag that defines the vertical or horizontal orientation of the path.

The velocities of the pool and of the atmosphere in a flow path may be forced to be equal rather than calculated independently.

The partition of the flow area between pool and atmosphere (defined by the flow path void fraction) is ordinarily based on current geometry. Alternatively, it may be defined such that either pool or atmosphere is preferentially transmitted. These options, referred to as pool-first and atmosphere-first, respectively, should be used with great care (if at all), as they often produce unexpected results. The preferential flow is accomplished by overriding the normal definition of the flow path void fraction if the preferred phase is available to flow through the flow path, setting it to 0.0 for a pool-first path and 1.0 for an atmosphere-first path. If the preferred phase is not available within the junction opening, the other phase is permitted to flow in the normal manner.

1.6.7 Momentum Flux Terms

Momentum flux terms can be included as an option in certain designated flow paths, using information supplied by the user to calculate them. These terms may be important in paths where large flow accelerations may occur, when multidimensional flow patterns occur, or where frictional and form loss terms are very small. In order to obtain a meaningful velocity difference for the momentum flux terms in the direction of flow, the user is required to specify appropriate upstream and downstream flow areas. As currently implemented, the momentum flux terms are calculated in a numerically explicit manner and are only one-dimensional in nature; i.e., they do not currently include the cross-derivative term $v_y \left(\partial v_x / \partial y\right)$, where y is perpendicular to x, to the direction of flow. The current model treatment will, therefore, be incomplete in situations where velocity gradients transverse to the direction of flow are important.

Although the momentum flux terms can be included, other terms in the CVH equations on the order of the velocity squared (the kinetic energy terms) are still omitted.

1.6.8 Flow Blockage Controlled by Other Packages

Phenomena modeled by other packages in MELCOR can result in changes to the geometry—area and flow resistance—of flow paths modeled by the MELCOR hydrodynamic packages. Although such effects can be captured through use of control functions to define effective valves and friction coefficients, this approach places a large burden on the user. Among the most important of these for reactor safety calculations are changes resulting from degradation of the reactor core that result in the blockage of flow by relocation of core debris and/or the opening of flow paths by failure of the channel boxes in a BWR or of the core shroud (baffle) in a PWR. In MELCOR, these effects can be easily included in the modeling of a flow path. The user is required only to specify the flow orientation and the core cell or cells that contribute to the geometry of the path. When particulate debris is involved, the calculation is based on the Ergun equation for flow in a porous medium.

1.6.9 Flashing of Superheated Pool Flows

An option is available for flow paths to account for flashing. It is off by default and can be enabled independently for forward and/or reverse flow in each flow path. If it is enabled, the flow of pool material will be partitioned between liquid and vapor, based on equilibration (and stagnation) at the pressure of the receiving volume.

The partition between liquid and vapor is calculated from the average enthalpy, and a fraction of the liquid is assigned to the fog field. By default, this is taken as the fraction of a Rosin-Rammler distribution that lies below a maximum diameter. If the RN1 package is active, the cut-off diameter is taken as the maximum aerosol size treated by MAEROS (DMAX on record RN1_ASP, with a default of 50μm). If the RN1 package is not active, the maximum size is defined by a sensitivity coefficient with the same default value, 50μm. The user may also specify the fraction directly through a sensitivity coefficient. By default, the Sauter mean diameter that characterizes the Rosin-Rammler distribution is defined by a sensitivity coefficient with a default value of 65μm. The user can chose to replace this default, on a case-by-case basis, with either a different constant value or the value of a REAL control function. If, after equilibration, the mass of fog in any volume exceeds the maximum permitted, the excess is added to the pool of that volume.

If the MELCOR RN package is active, water droplets in the atmosphere are considered to behave as aerosols. In this case, the Rosin-Rammler distribution is ordinarily used to distribute the new fog over the water aerosols sections (size bins). A user option switch is provided to defeat this, in which case the RN1 package will calculate the distribution based on the assumption that the new fog resulted from condensation. However, the resulting average droplet diameter is typically far too small, and the option should be used only for purposes of testing and demonstration.

1.6.10 New Stratified Counter Current Flow Model

This model is intended to model stratified counter current flow of gases in a flow path. User input is available to allow coupling of flows in two paths through momentum exchange, using a model that is conceptually based on stratified flow, and produces counter-current flow behavior that nearly duplicates the Epstein-Kenton correlations.

The model couples only atmospheric flows, but is not limited to cases where there are no pools or pool flows present. It should be useful for calculations including those involving natural circulation in PWR hot legs. If the coupled paths do not connect to the same pair of volumes (i.e., common in PWR hot leg modeling), the user must specify the two volumes that define atmosphere densities for use in the Froude Number.

Note that there is no control function or plot variables associated for this input.

1.7 Flow Paths to and from Core Volumes

The core package employs a model, referred to as the "dT/dz" model, to infer an axial temperature variation in a control volume containing core cells. (The details of this model are presented in the COR Package Reference Manual.) It uses the mass flow into the bottom of the control volume as a boundary condition and, as currently coded,

assumes that fluid flows up through the core. It further assumes that this corresponds to the direction of positive flow, which places a constraint on the definition of flow path orientations: the user must define the "*To*" and "*From*" control volumes for core flow paths such that the sign of the velocity is positive when fluid flows up through the core. This is illustrated in Figure 1.6.

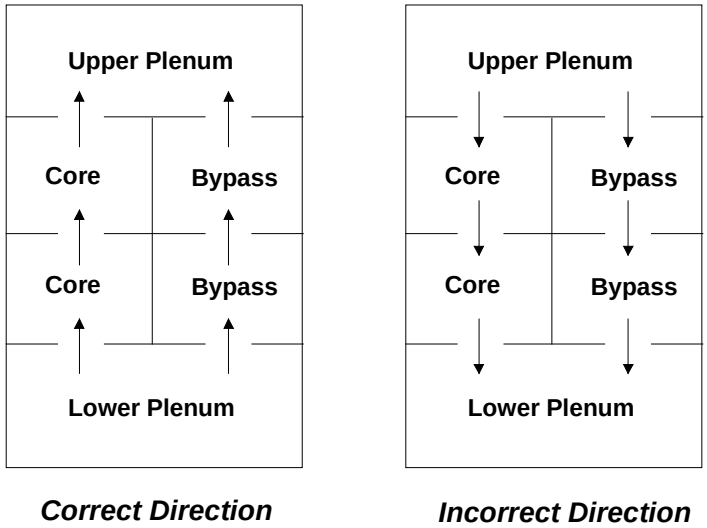


Figure 1.6 Flow Paths for "*To*" and "*From*" Core Volumes

1.8 Modeling Breaks and Failures

All types of failures which lead to the opening of additional paths for fluid flow are modeled by using flow paths containing valves which are defined to open when a failure criterion is reached. This includes pipe breaks, melt-through or catastrophic failure of the pressure vessel, failure of rupture disks and blowout panels, and failure of containment. Valve models are based on control functions, as described in Section 1.6.1. The flexibility of the MELCOR CF Package, described in the CF Package Users' Guide, makes it possible to define extremely general failure criteria through user input to the CF package in addition to simple pressure or temperature thresholds.

Input to the CF package includes user-specified bounds on the permitted values of the control functions. These limits may be changed on restart of MELCOR. This makes it possible to include in FL input all the failure paths and failure criteria that will be required to investigate a variety of scenarios. In any given MELCOR run, all but one can be inactivated by limiting the unwanted failure areas to zero through limits on the associated control functions.

For related scenarios, the transient response of the system will be identical until failures occur. If restart dumps are generated at the times when failures occur, the invariant portions of the calculation need not be repeated. Multiple branches may be followed by restarting MELCOR from the common point with one or another of the failure modes enabled by appropriate modification of control function bounds through CF package input.

1.9 MACCS Interface

The MACCS consequence code requires data on fluid flows and radionuclide transport to the environment through flow paths identified as release paths. Input records and plot variables allow one or more flow paths in the FL package to be identified as release paths for MACCS. (It is possible to define release as corresponding to negative flow in a path.) For all such paths, data about fluid and radionuclide transport through the path will be preserved in the database and written to the plot file. Masses are also available as control function arguments. This capability can also be used to obtain such data in plottable form for other applications.

1.9.1 MACCS Releases by Ring (or SFP Batch)

The MACCS code is able to distinguish mass released through a MACCS flow path by ring (or SFP batch). This is important in SFP analysis as each ring may represent fuel from different batches of fuel with significant differences in burn-up and activities. To strictly model such behavior it would be necessary to define unique classes for each MELCOR ring. However, an approximate method is available in MELCOR to accomplish this. The release of each RN class mass by ring is approximated by multiplying the fractional fuel release of each class in each ring by the fractional release of each class to the environment. The inherent approximation made is that decontamination factor (DF) following release from the fuel is the same for RN released from all rings, i.e., there is no distinction by ring. In reality, temporal release from each ring is different and will have different DFs dependent on condition in the containment at the time of release, though this approximation is adequate for most applications.

The release by batch is accomplished by first calculating the initial mass associated with each batch, for each RN class j, $FuelInv_{i,\mathrm{r}}$. Then for each RN class j, the total mass released from fuel for each ring is accumulated as the calculation progresses, $Mrel_{fuel,i,r}$. For each RN class, the fractional fuel released for each ring is then tracked, $Mrel_{fuel,i,r}/FuelInv_{i,\mathrm{r}}$.

Independently, the mass released to the environment through a MACCS flow path is tracked for each RN class, j, $Mrel_{env,i}$. Finally, the environmental batch release fraction is then calculated from:

$$\mathrm{Frel}_{\mathrm{env,i,r}} = \frac{\mathrm{Mrel}_{\mathrm{fuel,i,r}}}{\sum_{\mathrm{r}} \mathrm{Mrel}_{\mathrm{fuel,i,r}}} \times \frac{\mathrm{Mrel}_{\mathrm{env,i}}}{\boldsymbol{Fuel}\mathrm{Inv}_{i,\mathrm{i}}} \tag{1.9}$$

2 User Input

FL input to MELGEN defines the geometry of flow paths connecting control volumes, any initial velocities in these flow paths, and various options in the modeling. FL input to MELGEN also defines momentum sources (pumps) and control logic (valves) through reference to tabular and control functions defined in the TF and CF packages.

FL input to MELCOR is limited to modification of the momentum exchange length for two-phase flow. However, control function data may be modified on a restart, giving the

user significant access to control logic affecting pumps and valves. It is common to define input decks to include representations of several possible failures, with all but one disabled through limits imposed on values of control functions through CF package input. This allows several scenarios to be investigated by using modifications to CF input to branch calculations from intermediate restarts without rerunning the earlier, invariant portion of the transient.

2.1 MELGEN User Input

The user input for the FL package is described below. One set of the following records is required for each flow path. The number of flow paths is limited only by the available memory on the computer.

2.1.1 FL Records Associated With a Particular FL_ID record

The records described in this section are associated with a particular flow path identified on the 'FL_ID' record. The FL_ID record starts a block of input records that pertain to the flow path specified on that record. It is assumed that, until a new input block is encountered, subsequent records apply to the flow path specified on the FL_ID record. As an example, it is implied by the records in green that follow are applied to the 'CONTAIN LEAK' flow path and the record in red apply to the 'BASEMENT-DOME' flow path:

```
FL_ID 'CONTAIN LEAK'    1
FL_FT 'CV50-LOWER DOME' 'CV811-1ST_SOURCE' 7.5     7.5
FL_GEO    0.46E-05   1.0   1.0
FL_JSW  3      NoBubbleRise NoBubbleRise ! KFLGFL IBUBF IBUBT
FL_SEG   1
         1      0.46E-05   1.0   0.242E-02   0.5E-05 CONST     0.16E+02
FL_ID 'BASEMENT-DOME'    5
FL_FT 'CV5-BASEMENT' 'CV55-DOME' 19.08            19.08
FL_GEO    0.33860000E+03   0.19080000E+02   0.10000000E+01
FL_USL    2.0    2.0    1.0    1.0
FL_SEG   1
         1      3386.0   19.08   19.08   0.50000000E-05 CONST   16.0
```

FL_ID – Flow Path Name and Sequence Number

Required

This record defines the flow path name and a user defined sequence number. The second field IFPNUM is optional and may be omitted.

(1) FPNAME

User defined flow path name.

(type = character*16, default = none, units = dimensionless)

(2) IFPNUM

User defined flow path sequence number. If omitted, a sequence number will be assigned by incrementing the last sequence number

by 10.
Optional.

(type = integer, default = none, units = dimensionless)

Example

```
FL_ID 'INLET-BUNDLE 1' 20
```

FL_DELETE – Ignore all records associated with FL_ID

Optional

This record is processed after all records in the input file are read. It will turn off all FL records related to the FPNAME identifier.

(1) FPNAME
User-defined flow path name.

(type = character*16, default = none, units = none)

Example

```
FL_DELETE  'INLET-BUNDLE 1'
```

FL_FT – Control Volume Data that the Flow Path Connects

Required

This record defines the control volumes that the flow path connects. The flow is positive when it flows from the "*From*" volume to the "*To*" volume. The altitude of the connection with "*From*" and "*To*" control volumes is also defined.

(1) KCVFM
"*From*" control volume name.

(type = character*16, default = none, units = dimensionless)

(2) KCVTO
"*To*" control volume name.

(type = character*16, default = none, units = dimensionless)

(3) ZFM
Altitude of "*From*" junction.

(type = real, default = none, units = m)

(4) ZTO
Altitude of "*To*" junction.

(type = real, default = none, units = m)

Example

```
FL_FT   CVolume550   CVolume110   17.40   17.40
```

FL_GEO – Flow Path Geometry

Required

This record defines the flow path geometry. Only the first three fields need to be present. If record FL_JLF and/or FL_JLT are present, data on those records will define the junction opening, and the values of FLHGTF and/or FLHGTT, whether input or default, will not be used.

(1) FLARA

Flow path area.

(type = real, default = 1.0, units = m^2)

(2) FLLEN

Flow path length.

(type = real, default = 1.0, units = m)

(3) FLOPO

Fraction of flow path open.

(type = real, default = 1.0, units = dimensionless)

(4) FLHGTF

"*From*" junction flow path opening height.

(type = real, default = diameter of circle with area FLARA for a horizontal flow path = radius of circle with area FLARA for a vertical flow path, units = m)

(5) FLHGTT

"*To*" junction flow path opening height.

(type = real, default = diameter of circle with area FLARA for a horizontal flow path = radius of circle with area FLARA for a vertical flow path, units = m)

Note: The "*From*" and "*To*" junction flow path opening heights may be defined by the user. If the user changes the default for at least one variable, both heights must be input; if not, these fields may be absent.

Examples

```
FL_GEO   0.567   10.517   1.0
```

```
FL_GEO   0.706E-01   0.5   0.0   0.0   0.0
```

FL_JSW – Flow Path Junction Switches

Optional

This record defines the type of flow path. A flow path may be horizontal or vertical. The algorithm used to define the equivalent opening height is different for the two cases. A pool-first flow path will only flow pool water until all available pool water is exhausted. When there is no pool available, the flow path will flow atmosphere. The atmosphere-first flow path is equivalent to the pool-first flow path with the pool and atmosphere roles reversed. The normal horizontal or vertical flow paths will flow a mixture of pool and atmosphere consistent with the pool elevation, velocities, and flow path opening elevation and height. It is also possible to specify that the pool and atmosphere must have the same velocity.

A flow path may also be defined as a one-way path, allowing flow in one direction only, to simulate the behavior of an idealized check valve. This is done by adding a leading digit of 1 for forward-flow-only or 2 for reverse-flow-only to the variable KFLGFL that defines the type of flow path.

A bubble rise model may be specified that includes thermodynamic interactions and, if the RN package is active, RN scrubbing. The calculation of heat and mass transfer is performed for any acceptable non-zero value of IBUBF or IBUBT. The specific non-zero value is significant only if the RN package is active.

(1) KFLGFL

Flag for type of flow path.

(a) = 0
Normal vertical flow path.

(b) = 1
Atmosphere-first vertical flow path.

(c) = 2
Pool-first vertical flow path.

(d) = 3
Normal horizontal flow path.

(e) = 4
Atmosphere-first horizontal flow path.

(f) = 5
Pool-first horizontal flow path.

(g) = 6
Pool velocity = atmosphere velocity, vertical flow.

(h) = 7

Pool velocity = atmosphere velocity, horizontal flow.

(i) = 10 – 17
Same as 0 – 7, but only forward flow is permitted.

(j) = 20 – 27
Same as 0 – 7, but only reverse flow is permitted.

(type = integer, default = 0, units = dimensionless)

(2) IBUBF

"*From*" junction bubble rise model switch.

(a) 0 or NoBubbleRise
No bubble rise physics.

(b) 1 or AerosolIodineScrubbing
SPARC model with RN aerosol and iodine vapor scrubbing.

(c) 2 or AllScrubbing
SPARC model with RN aerosol and vapor scrubbing of all classes (same modeling as iodine vapor scrubbing)

(d) -1 or NoScrubbingRN
SPARC model with no RN scrubbing.

(e) -2 or AerosolScrubbing
SPARC model with scrubbing of RN aerosols only.

(f) -3 or IodineScrubbing
SPARC model with scrubbing of RN iodine vapor only.

(type = integer, default = 0, units = dimensionless)

(3) IBUBT

"*To*" junction bubble rise model switch.

(a) 0 or NoBubbleRise
No bubble rise physics.

(b) 1 or AerosolIodineScrubbing
SPARC model with RN aerosol and iodine vapor scrubbing.

(c) 2 or AllScrubbing
SPARC model with RN aerosol and vapor scrubbing of all classes (same modeling as iodine vapor scrubbing)

(d) -1 or NoScrubbingRN
SPARC model with no RN scrubbing.

(e) -2 or AerosolScrubbing

SPARC model with scrubbing of RN aerosols only.

(f) -3 or IodineScrubbing
SPARC model with scrubbing of RN iodine vapor only.

(type = integer, default = 0, units = dimensionless)

Example

```
FL_JSW   0   NoBubbleRise   AerosolIodineScrubbing
```

FL_USL – User Specified Loss Coefficients

Optional

The user can specify forward and reverse loss coefficients and choked flow discharge coefficients. If this record is input, then at least two fields must be input. The first two fields are interpreted as the loss coefficients. The third and fourth fields are the forward and reverse choked flow discharge coefficients.

The loss coefficients determine the form loss pressure drop across a flow path; the forward coefficient is used when the velocity is positive and the reverse coefficient is used when the velocity is negative. See Section 1.2.1 of this users' guide and Section 5.4 in the CVH/FL Reference Manual for additional information.

In the test for choking, critical flow is determined from a correlation and then multiplied by a discharge coefficient. The coefficient may be different for forward flow (positive velocity) and reverse flow (negative velocity). A minimum velocity, below which the test is bypassed, can be defined by sensitivity coefficient C4402(1).

(1) FRICFO
Forward loss coefficient.

(type = real, default = 1.0, units = dimensionless)

(2) FRICRO
Reverse loss coefficient.

(type = real, default = 1.0, units = dimensionless)

(3) CDCHKF
Choked flow forward discharge coefficient.

(type = real, default = 1.0, units = dimensionless)

(4) CDCHKR
Choked flow reverse discharge coefficient.

(type = real, default = 1.0, units = dimensionless)

Example

```
FL_USL  6.00   6.0  1.0   1.0
```

FL_VEL – Initial Atmosphere and Pool Velocities

Optional

The initial pool and atmosphere velocities may be defined by the user. If this record is present, both velocities must be input; if it is absent, both initial velocities will be taken as 0.0.

(1) VLFLAO

Atmosphere velocity.

(type = real, default = 0.0, units = m/s)

(2) VLFLPO

Pool velocity.

(type = real, default = 0.0, units = m/s)

Example

```
FL_VEL   10.0   0.0
```

FL_LME – Length for Pool/Atmosphere Momentum Exchange

Optional

This record defines the length over which momentum exchange occurs when pool and atmosphere flows share the flow path. The value used affects entrainment and flooding by controlling the balance between the resulting force between the two flows (which tends to couple them and reduce slip) and buoyancy (which tends to separate pool from atmosphere). A larger value tends to promote coupling between pool and atmosphere while a smaller one tends to enhance their separation.

The default for horizontal flow paths is the inertial length defined on record FL_GEO. The default for vertical flow paths is the distance between the lowest point in the flow path (including the junction opening) and the highest point, as defined by the junction elevations on record FL_FT and the opening heights from record FL_GEO (or default). See Section 1.3.2 of this users' guide and Section 5.5 of the CVH/FL Reference Manual for more information.

(1) XL2PF

Length for pool/atmosphere momentum exchange.

(type = real, default as discussed above, units = m)

Example

```
FL_LME   0.1
```

FL_EDF – Flow Path Connection to EDF

Optional

This record directs that the cumulative flows of masses and enthalpies through the flow path be recorded to an external data file. The resulting file can then be used to define mass and energy sources to the CHV package in a subsequent calculation.

If a file is written, each record will contain NUMMAT cumulative masses, the cumulative flow of enthalpy associated with pool material, and the cumulative flow of enthalpy associated with atmosphere materials. Enthalpies are defined with respect to the normal MELCOR reference points.

(1) IP2EDF

User-assigned name of associated External Data File (EDF) file.

(type = character*16, default = none, units = dimensionless)

Note: IP2EDF must be the name of a valid "PUSH" file containing exactly NUMMAT+2 channels. See the EDF Users' Guide for further input requirements, including filenames and record frequencies.

Example

```
FL_EDF    EDF35
```

FL_JLF – Junction Limits for "*From*" Volume

Optional

This record defines the junction opening for the "*From*" volume by directly specifying the range of junction elevations for that volume. If this record is present, the value of *FLHGTF* on record FL_GEO, whether input or default, will not be used.

(1) ZBJFM

Elevation of bottom of junction opening for the "*From*" volume. Must lie between bottom of "*From*" volume and nominal junction elevation (inclusive).

(type = real, default = none, units = m)

(2) ZTJFM

Elevation of top of junction opening for the "*From*" volume. Must lie between nominal junction elevation and top of "*From*" volume (inclusive).

(type = real, default = none, units = m)

Note: If this record is present, both fields must be input.

Example

```
FL_JLF   2.0   2.1
```

FL_JLT – Junction Limits for "*To*" volume

Optional

This record defines the junction opening for the "*To*" volume by directly specifying the range of junction elevations for that volume. If this record is present, the value of *FLHGTT* from record FL_GEO, whether input or default, will not be used.

(1) ZBJTO

Elevation of bottom of junction opening for the "*To*" volume. Must lie between bottom of "*To*" volume and nominal junction elevation (inclusive).

(type = real, default = none, units = m)

(2) ZTJTO

Elevation of top of junction opening for the "*To*" volume. Must lie between nominal junction elevation and top of "*To*" volume (inclusive).

(type = real, default = none, units = m)

Note: If this record is present, both fields must be input.

Example

```
FL_JLT   1.9   2.0
```

FL_BLK – Data for Blockage of Flow by Another Package

Optional

Blockage of flow in response to change of geometry in another package (reduction of flow area or redefinition of friction) will be calculated if data are entered on this record. If a dataset is entered, no other control of the flow area is possible, and

inclusion of a valve (FL_VLV) record is not permitted. Only one blockage dataset may be entered for a flow path.

For the time present, COR is the only package that can define blockage of flow. The COR flow blockage model can automatically adjust the area and flow resistance of specified flow paths to include the effects of blockage by core debris as modeled by the COR package. It can also model the opening of a flow area between the channel and bypass of a BWR when the separating channel box fails.

The user must specify which core cells are to be associated with the flow path; the additional data required are defined below. Fields 1–5 are required; field 6 is optional.

(1) OPTION

Flow geometry to be modeled in this path.

(a) AXIAL

Axial flow geometry. May be used for a BWR only if the channel and bypass are not distinguished in any of the core cells involved.

(b) AXIAL-C

Axial flow geometry. Used for a BWR or the peripheral region of a PWR.

(c) AXIAL-B

Axial flow geometry, considering only the bypass region for a BWR.

(d) RADIAL

Radial flow geometry. Not recommended for a BWR because the channel and bypass regions are treated as combined

(e) RADIAL-C

Radial flow geometry, considering only the channel region. Used for paths connecting to the peripheral region of a PWR.

(f) RADIAL-B

Radial flow geometry, considering only the bypass region for a BWR.

(g) CHANNEL-BOX

Connection between channel and bypass of a BWR that opens when the channel box fails.

(h) CORE-SHROUD

Connection between channel and bypass in the peripheral region of a PWR that opens when the core shroud (baffle) fails.

(i) PBR-R
Specifies pebble bed friction factor for radial flow in PBR. Note that there is no actual blockage, but the pebble bed friction factors will be used.

(j) PBR-A
Specifies pebble bed friction factor for axial flow in PBR.

(type = character*12, default = none, units = dimensionless)

(2) ICORCR1

(3) ICORCR2

Define the limiting core rings to be associated with the flow by defining two corner core cells. Each core cell in the range must be associated with either the "From" or the "To" control volume for the flow path as defined on the FL_FT record.

(type = integer, default = none, units = dimensionless)

(4) ICORCA1

(5) ICORCA2

Define the limiting core axial levels to be associated with the flow path by defining two corner core cells. Each core cell in the range must be associated with either the "From" or the "To" control volume for the flow path as defined on the FL_FT record.

(type = integer, default = none, units = dimensionless)

(6) FLMPTY

Form loss coefficient for empty geometry to be added to the value from the Ergun correlation.
Optional.

(type = real, default = 1.0, units = dimensionless)

Example

```
FL_BLK   AXIAL-C   1   1   7   7   1.0
```

FL_SEG – Piping Segment Parameters

Required

The user must specify several parameters for each segment; at least one segment is required. Some of the parameters are optional. Each record must contain all the data for a segment; the data for a segment cannot be split across records.

(1) IPNSG

User-defined number of segments.

(type = integer, default = 0, units = none)

The following data are input as a table with length IPNSG:

(1) NSEG

Piping segment index is used for sequencing.

(type = integer, default = none, units = none)

(2) SAREA

Segment flow area.

(type = real, default = none, units = m^2)

(3) SLEN

Segment length.

(type = real, default = none, units = m)

(4) SHYD

Segment hydraulic diameter. The conventional definition is given by 4 times the flow area divided by the wetted perimeter.

(type = real, default = none, units = m)

(5) SRGH

Surface roughness.
Optional.

(type = real, default = 5.E-5, units = m)

(6) LAMFLG

Key setting one of the following values:

(a) 0 or CONST
(b) -1 or CF

Optional.

(type = integer/character*6, default = 0/CONST, units = none)

If LAMFLG = CONST

(7) SLAM

Laminar flow coefficient.
Optional.

(type = real, default = value of sensitivity coefficient C4404(13), whose default value is 16.0, units = dimensionless)

If LAMFLG = CF

(7) CFNAME

Control function name, which is used to define the laminar flow coefficient.

(type = character*16, default = none, units = dimensionless)

Note:

1. The field LAMFLG is optional. The default for this key is 0/CONST. If SLAM and CFNAME fields are absent, the default for the laminar flow coefficient will be equal to the value of sensitivity coefficient C4404(13), or 16.0.

2. Field SRGH can be absent only if the fields SLAM and CFNAME are absent.

Example

```
FL_SEG 2 !NSEG  SAREA SLEN SHYD        SRGH  LAMFLG  SLAM/CFNAME
           1    3.189 1.675 3.490     0.1E-04 CONST  0.0
           2    1.391 2.185 1.060E-02 0.1E-04 CONST  0.0
```

2.1.2 FL Records Not Associated With a Particular FL_ID record

The following records are general records and do not require an FL_ID record.

FL_FLSH – Flow Path Flashing for Forward (Positive) and Reverse (Negative) Flows

Optional

This record defines parameters for flashing of forward (positive) flows, into the "*To*" volumes, and for flashing of reverse (negative) flows, into the "*From*" volumes. For FORWARD flows listed on this record, flashing model will be activated to distribute pool flow among pool, fog, and vapor in the "*To*" volume. Fog will also be added to the water-class aerosol if the RN1 package is active. For all other flow paths no flashing calculation will be done. Pool flow into the "*To*" volume will be added directly to the pool in that volume. The same is also right for REVERSE flows.

(1) NFLSH

Number of flow paths.

(type = integer, default = none, units = none)

The following data are input as a table with length NFLSH:

(1) NFL

Table row index.

(type = integer, default = none, units = none)

(2) FLNAME

Flow path name.

(type = character*16, default = none, units = none)

(3) FLSHDIR

Direction of flow flashing.

(a) FORWARD
(b) REVERSE

(type = character*7, default = none, units = none)

(4) ISOPT

Options for defining the Sauter mean diameter of water droplets.

(a) -1 or CF
Use REAL control function CFNAME (input below) whose value defines the Sauter mean diameter.

(b) 1 or SC
Use value of sensitivity coefficient C4500(3) (no additional field is required).

(c) 0 or CONST
Use constant value SAUTER input below.

(type = character*16, default = none, units = none)

optionally followed by

(5) CFNAME

Name of REAL control function whose value defines the Sauter mean diameter of water droplets.

(type = character*16, default = none, units = none)

or

(5) DIAM

Sauter mean diameter of water droplets.

(type = real, default = none, units = none)

The last field is optional and may be omitted:

(6) IFOGOPT

Fog option flag.

(a) =0
Distribute fog over RN sections directly.

(b) $\neq 0$
Allow RN to calculate distribution, assuming condensation.

(type = integer, default = 0, units = none)

Example

```
FL_FLSH  3 ! NFL FLNAME FLSHDIR SC/CF/CONST CFNAME/DIAM  IFOGOPT
             1   FL31   FORWARD SC
             2   FL199  REVERSE CF          CF199        0
             3   FL230  FORWARD CONST              2.5   1
```

FL_VLV – Valve Input

Optional

Only one valve may be input for a given flow path.

Valves may be used to open and close flow paths during the course of a calculation. The flow path area may take on any value between 0 (fully closed) and *FLARA* (fully open). The fraction open is defined by one or more control functions. It is defined directly by a single control function if no trip is used (although three fields must still be present on the record).

If a trip is specified, a trip control function is used to define "*on-forward*", "*off*" and "*on-reverse*" states. RELAP and TRAC users take note: A MELCOR trip is not what you are used to. Different control functions are used to define the open fraction for "*on-forward*" and for "*on-reverse*" states. If the trip is "*off*", the valve opening remains unchanged from the previous timestep. See Section 1.6.1 of this document and the CF Users' Guide for more information.

(1) NVALVE

Number of valves.

(type = integer, default = none, units = none)

The following data are input as a table with length NVALVE:

(1) NV

Table row index.

(type = integer, default = none, units = none)

(2) VLVNAME

Name of valve.

(type = character*16, default = none, units = none)

(3) FLNAME

Name of connected flow path.

(type = character*16, default = none, units = none)

(4) KEYTRIP

Key setting one of the following values.

(a) 0 or NoTRIP

A trip is not used and the fraction open is defined by the "*on-forward*" control function.

(b) 1 or UseTRIP

A trip is used, and the fraction open of the flow path for "*on-forward*" and "*on-reverse*" states of the trip is defined by the "*on-forward*" and "*on-reverse*" control function names.

(c) NoTRIPCV

Indicates that this valve is described by a valve loss coefficient, C_V (see below).

(type = integer, default = none, units = none)

if KEYTRIP = 0/NoTrip

(5) NVFONF

No trip is defined. This control function defines the fraction open.

(type = character*16, default = none, units = dimensionless)

or if KEYTRIP = 1/UseTrip

(5) NVFONF

Control function name used to define the fraction open of the flow path for an "*on-forward*" state of the trip.

(type = character*16, default = none, units = dimensionless)

(6) NVFONR

Control function name used to define the fraction open of the flow path for an "*on-reverse*" state of the trip.

(type = character*16, default = none, units = dimensionless)

(7) NVTRIP

Trip control function name.

(type = character*16, default = none, units = dimensionless)

A special valve type has been introduced into MELCOR to indicate the valve is modeled with a valve flow loss coefficient, C_V. To utilize this model, the user indicates that the valve is a 'NoTRIPCV' for field KEYTRIP, and then supplies a CF for specifying the value of C_V for the valve. Currently, the valve must be on a single segment flowpath and takes the pipe diameter from this segment. Customary engineering units expected for this parameter are gpm/sqrt(psi) and are from the definition of C_V.

For such a valve, MELCOR does not perform a wall-friction loss calculation and does not use any user-supplied loss coefficient (K), even though the user may specify loss coefficients on the FL_USL record. Internally, the code will convert the C_V value from the control function to a dynamically calculated loss coefficient, K. The user can still specify multiple segments for this flow path, though MELCOR will take the pipe diameter from the first segment and will disregard the surface roughness, and thereby ignore the wall friction loss.

Example

```
FL_VLV  6 ! NV VLVNAME FLNAME KEYTRIP NVFONF  *NVFONR   NVTRIP
             1 Valve1   FL31   NoTRIP  CF130
             2 Valve2   FL199  NoTRIP  CF199
             3 Valve3   FL230  NoTRIP  CF232
             4 Valve4   FL398  UseTRIP CF502    CF502    CF501
             5 Valve5   FL399  UseTRIP CF402    CF402    CF401
             6'TestValve' 'VALVE'  NoTRIPCV  'CVvsTime'  !This is a valve describe by a Cv
```

FL_PMP – Pump Input Data

Optional

Only one pump may be input for a given flow path.

(1) NPUMP

Number of pumps.

(type = integer, default = none, units = none)

The following data are input as a table with length NPUMP:

(1) NP

Table row index.

(type = integer, default = none, units = none)

(2) PNAME

Name of the pump.

(type = character*16, default = none, units = dimensionless)

(3) FLNAME

Name of the flow path connected to this pump.

(type = character*16, default = none, units = dimensionless)

(4) PTYPE

Pump type.

(a) FANA

The FANA pump model represents a simple fan in its normal operating range. However, by an appropriate

choice of parameters, the FANA model can be used to approximate a constant-velocity coolant pump. The input parameters are illustrated in the CVH/FL Reference Manual. Coupling to the momentum equation is explicit in time, and instabilities may arise for large timesteps. The problem may usually be mitigated by increasing the inertial length of the flow path, FLLEN, on record FL_GEO.

Note: FANA is valid ONLY for flow paths of type KFLGFL = 6, 7, 16, 17, 26 or 27(see record **FL_JSW**). MELGEN will abort for other types of flow paths.

(b) QUICK-CF

The QUICK-CF pump type allows the user to define the pump head through a control function. This function might be as simple as a tabular function of velocity or time. The complexity of the model is limited only by the ingenuity and patience of the user; an example of a (partial) homologous pump model is shown in the CF User's Guide. Coupling to the momentum equation is explicit in time, and instabilities may arise for large timesteps. The problem may usually be mitigated by increasing the inertial length of the flow path, FLLEN, on record FL_GEO.

(type = character, default = none, units = dimensionless)

The additional data required to define a pump depend on the pump type. These data are interpreted according to the pump type flag.

If PTYPE = FANA

(5) DPMAX

Maximum pressure head.

(type = real, default = none, units = Pa)

(6) VPZERO

Volumetric flow rate at zero pressure head. The pressure head is zero for volumetric flow rates greater than or equal to VPZERO.

(type = real, default = none, units = m^3/s)

(7) VPMAX

Volumetric flow rate at maximum pressure head. The pressure head is also set to DPMAX for flows less than VPMAX.

(type = real, default = 0, units = m^3/s)

(8) ITRIP

Pump trip flag.

(a) 0 or ON

The pump is always on; otherwise, the pump is on and the pressure head is calculated only if the value of a trip is nonzero.

(b) -1 or CF

The value of control function is used for the trip value.

(c) 1 or TF

The tabular function is used for the trip value. The tabular function independent argument is time.

(type = integer / character*2, default = 0, units = dimensionless)

If ITRIP = -1 or CF (Pump(i).ITRIP == 1) then

(9) CFNAME

The name of control function used for the trip value.

(type = character*16, default = none, units = dimensionless)

or if ITRIP = 1 or TF (Pump(i).ITRIP == -1) then

(9) TFNAME

The name of tabular function used for the trip value.

(type = character*16, default = none, units = dimensionless)

If PTYPE = QUICK-CF

(5) IPUMCF

Control function name defining the pressure head; the values from this control function should have units of Pascals.

(type = character*16, default = none, units = dimensionless)

Examples:

```
FL_PMP 5 !NP PNAME FLNAME PTYPE DPMAX VPZ VPM ITR CFNAME/TFNAME
         1  Pump1  FL280 QUICK-CF                 CF266
         2  Pump2  FL203  FANA   5.0   2.
         3  Pump3  FL209  FANA   5.0   2.  5. ON
         4  Pump4  FL213  FANA   5.0   2.  5. CF   CF101
         5  Pump5  FL203  FANA   5.0   2.  5. TF          TF282
```

FL_VTM – Time Dependent Flow Path

Optional

The velocity through the flow path may be defined using tabular or control functions. The pool and atmosphere velocities are identical. Only one time dependent flow path dataset may be entered for a flow path.

(1) NVOFT

Number of time dependent flow paths.

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NVOFT:

(1) NFLT

Table row index.

(type = integer, default = none, units = none)

(2) FLNAME

Name of flow path that will be time dependent.

(type = character, default = none, units = dimensionless)

(3) NTFLAG

Time dependent flow path type flag.

(a) 1 or TF

Use tabular function name NFUN to define velocity versus time

(b) -1 or CF

Use control function name NFUN to define velocity versus time.

(type = integer, default = none, units = dimensionless)

(4) NFUN

Tabular or control function name to define the velocity versus time. The interpretation is dependent on the value of NTFLAG.

(type = character*16, default = none, units = dimensionless)

Example

```
FL_VTM   2 !NFLT FLNAME          NTFLAG NFUN
            1    FlowPath190     TF     TF19
            2    FlowPath700     CF     CF700
```

FL_MACCS – MACCS Release Paths Definition

Optional

This record allows identification of flow paths that serve as release paths for the consequences code MACCS. For these paths—and these paths only—data about

fluid and radionuclide transport through the path will be written to the plot file. (The record can, of course, also be used simply to force such data to be written to the plot file for other applications.)

(1) MACCSN

The number of MACCS release paths.

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length MACCSN:

(1) NFL

Table row index.

(type = integer, default = none, units = none)

(2) MACCSname

Name of the MACCS flow path. Note that you can specify a different MACCS flow path for each direction of flow.

(type = character*16, default = none, units = none)

(3) MACCSnum

Number for MACCS flow path (used in the binary plot file)

(type = integer, default = none, units = none)

(4) MCCSFP

The name of the flow path.

(type = character*16, default = none, units = none)

(5) DIRFL

Direction of flow corresponding to release:

(a) FROM or POSITIVE is in the direction from the 'FROM' CV to the 'TO' CV.
(b) TO or NEGATIVE is in the direction from the 'TO' CV to the 'FROM' CV.

(type = character*16, default = none, units = none)

(6) DF

The MACCS flow path was originally intended to track airborne release through a flow path to the environment providing input to the MACCS code. For this purpose, radionuclides that are trapped in filters, scrubbed in pools, or transferred by pool flowing through the flow path are not tracked in the MACCS flow path by default. However, this feature can also used to more generally track radionuclides between volumes such as in calculating decontamination factors. For such cases, all radionuclides entering the flow path, including those trapped on filters or scrubbed, or transported with flowing pool are tracked by the MACCS flow path.

(a) 'MACCS' or '-' indicates a standard MACCS flow path. These flow paths are read by MELMACCS as source for the MACCS code.
(b) 'DF' indicates that this flow path is used in calculating a DF. These flowpaths are ignored by MACCS

(type = character*16, default = MACCS, units = none)

Example

```
FL_MACCS   2 !NFL  MACCSnam    MACCSn  MCCSFP       DIRFL
               1    Leakage     110   FlowPath190    FROM
               2     Break      120   FlowPath700    TO
```

FL_MACCS_SFP – MACCS Spent Fuel Pool Definition
Optional

This record allows the user to specify information needed to create a radionuclide inventory release profile from the Spent Fuel Pool (see section 1.9.1 for additional information). This profile allows the user to track the release based on the 'radial ring' or spent fuel batch (i.e., age of the spent fuel out of reactor and burnup). Note this option requires the user to activate the FL_MACCS as needed. If not, an error will be issued.

(1) FCOMP2EL

Flag to add the compound mass to the elemental class (i.e, Cs in CsI and in CsM classes will be added to Cs class, I2 in CsI will be added to I2 class, while Mo in CsM will be added to Mo class). The elemental fractions were given in the sensitivity coefficients of the RN1 package. However, they are reported as separate new classes, which will not alter the values in the default or defined classes in the RN package. Default is FCOMP2EL = 0, which does not create these new classes. If FCOMP2EL = 1, the compound classes and elemental classes must be given, and the elemental fraction must also be provided. Note if FCOMP2EL = 1, then these extra 3 classes will be created in addition to NRNCLASS. Note that these new classes are only created for certain plot variables, and do not affect the calculations.

(type = integer, default = 0, units = none)

(2) RFCLFLAG

Flag to use internally code calculated class mass fraction or mass for each ring (RFCLFLAG = 0) or user input (RFCLFLAG = 1). If RFCLFLAG = 1, FL_MACCS_SFP3 record is required.

(type = integer, default = 0, units = none)

(3) NCVHFLAG

Typically, all COR cells in a ring would be associated with a batch release. However, the user may specify a range of COR cells, possibly overlapping multiple rings, that will be associated with a

batch on record FL_MACCS_SFP1. This range is identified by the user by specifying CVH volumes that are associated with those COR cells. This flag identifies whether the CVHNAME entered on FL_MACCS_SFP1 is for CV name or CV type. Default is CV name = 0. NCVHFLAG =1, it is CV type.

(type = integer, default = 0, units = none)

FL_MACCS_SFP1 - MACCS Spent Fuel Pool Definition 1.

Optional.

By default, MELCOR will assume that fuel from all COR cells in a ring are associated with a batch. However, with this table it is possible for the user to specify a set of CVH control volume(s) which are associated with COR cells for a batch. The user can either specify CV names or ICVTYP as indicated by field NCVHFLAG on record FL_MACCS_SFP.

(1) NBATCH

Number of batches(s) associated with the MACCS_SFP release. This number cannot exceed the number of rings in the model

(type = integer, default = 1, units = none)

The following data are input as a table with length NBATCH:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) BATCHNAME

Name of the ring being modeled.

(type = character*32, default = none, units = none)

(3) NCVH

Number of CVH volumes or CVH types associated with this ring for tracking radionuclide release from fuel. Maximum value for NCVH is expected to be 17 because MELCOR 2.1 has a limit of 17 character arrays which can be defined by 'MaxInputFields' for a given line. It is encouraged to use CV type input to minimize the input fields for a given line. If NCVHFLAG equals to 0, it is expected that the CVHNAME is a CV name. If NCVHFLAG equals to 1, it is expected that the CVHNAME is a CV type.

(type = integer, default = 1, units = none)

The following data are input as a row with length NCVH:

(*) CVHNAME

Name of CVH volume or ICVTYP. It is expected that the number of this entry to be NCVH.

(type = character*16, default = none, units = none)

FL_MACCS_SFP2 - MACCS Spent Fuel Pool Definition 2

Required if FL_MACCS_SFP is used.

Identification of CVTYPE for the environment release. In most cases, it should be the "to" volumes identified in the flow paths in FL_MACCS record. However, this is not always the case.

(1) NENV

Number of CV types associated with the environment release.

(type = integer, default = none, units = none)

The following data are input as a table with length NENV:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) CVTYPE

Name of control volume type.

(type = character*16, default = none, units = none)

FL_MACCS_SFP3 - MACCS Spent Fuel Pool Definition 3

Optional (Required only if RFCLFLAG on FL_MACCS_SFP is 1)

Specification of the class initial mass fraction or actual initial mass. Thus the number of records required is number of classes and the number of fields on each record is NBATCH.

(1) FLAGMOF

Flag for the input – FLAGMOF = 0 for fractional value, and FLAGMOF = 1 for mass value. Note if fractional release is desired, the MELCOR calculated class initial mass in the COR package is used. If mass value is entered, the environmental release mass for each ring and class will be calculated based on the input mass value.

(type = integer, default = 0, units = none)

The following data are input as a table with row length of NRNCLASS and column length of NBATCH:

(1) CLASSNO

Class number index. It should start from 1 to NRNCLASS

(type = integer, default = none, units = none)

(*) FRNMASS

Fractional ($F_{i,r}$ in Equation [1]) or mass value ($F_{i,r} \times Inv_{fuel,i}$ or $Inv_{fuel,i,r}$ in Equation [1]) for each class i and each ring r. A total of NRNCLASS times NBATCH is expected. See FLAGMOF for the definition of the input value. (*) signifies the number of columns must correspond to NBATCH.

(type = real, default = none, units = none)

Example

```
This example has 5 flow paths.  All MACCS flow paths connected to any CV volume designated
as type 'ENVIRON' is used to determine environment release.
FL_MACCS    5 !(
         !  NFL   MACCSName             MACCSnum MCCSFP             DIRFL
            1     'RACK_OVERFLOW_ENV_1'   901   'POOL_OVERFLOW_1' FROM
            2     'CASK_OVERFLOW_ENV_2'   902   'POOL_OVERFLOW_2' FROM
            3     'REF_BAY_LEAKAGE'       903   'LEAK-REFBAY'     FROM
            4     'REF_BAY_OVERFLOW'      904   'BO-REFBAY-ENV'   FROM
            5     'CASK_LEAKAGE'          905   'Leak'            FROM
FL_MACCS_SFP   1
FL_MACCS_SFP2  1
!             ENV#   CVTYPE
               1     'ENVIRON'
```

FL_MFX – Momentum Flux Input Data

Optional

Momentum flux will be calculated for the flow path if data are entered on this record.

(1) XCVAFM

Upstream ("*From*") control volume flow area appropriate for flow path (i.e., in the direction of flow).

(type = real, default = none, units = m^2)

(2) XCVATO

Downstream ("*To*") control volume flow area appropriate for flow path (i.e., in the direction of flow).

(type = real, default = none, units = m^2)

(3) NFLFM

Upstream flow path name.
Optional.
If absent or set to NO, an upstream flow velocity of zero will be used to calculate momentum flux.

(type = character*16, default = none, units = dimensionless)

(4) NFLTO

Downstream flow path name.
Optional.
If absent or set to NO, a downstream flow velocity of zero will be used to calculate momentum flux.

(type = character*16, default = none, units = dimensionless)

Example

```
FL_MFX   0.1389E-01   0.49327E-02   NO   FL210
```

FL_MCH – Mechanical model

Optional

This model applies pressure changes to a flow path similar to a turbine or compressor. Note DELP can be input through a control function argument, in addition to a constant value. When enters as a real value, it is a constant value option. When enters as a character, which must be corresponding to the name of the CF for this entry. The control function information will be used for this data.

(1) NUMMCH

The number of mechanical models

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUMMCH:

(1) NM

Table row index.

(type = integer, default = none, units = none)

(2) MCHNAME

The name of the mechanical model.

(type = character*16, default = none, units = none)

(3) MCHNUM

The mechanical model sequence number

(type = integer, default = none, units = none)

(4) MCHTYP

The type of the mechanical model (not used).

(a) PRESSURE
(b) DELTA_H

(type = character*16, default = none, units = none)

(5) FPNAME

Name of flow path through model.

(type = character*16, default = none, units = none)

(6) DELTF

Forward temperature increase (decrease).

(type = real, default = none, units = K)

(7) DELTR

Reverse temperature increase (decrease).

(type = real, default = none, units = K)

(8) DELP

Pressure boost in forward direction (constant value or control function name)

(type = real or character*16, default = none, units = Pa)

(9) EFFIC

Efficiency.
Optional.

(type = real, default = 1.0, units = none)

Example

```
FL_MCH   1
             1    Turbine 10 none FL100  100.0 50.0 1.0e5 0.88
             1    Turbine 20 none FL101..100.0 50.0 CFTDP 0.88
```

FL_IHX – Heat Exchanger Model

Optional

This model simulates a heat exchanger using two flow paths.

(1) NUMIHX

The number of heat exchanger models

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUMIHX:

(1) NX

Table row index.

(type = integer, default = none, units = none)

(2) IHXNAME

The name of the heat exchanger model.

(type = character*16, default = none, units = none)

(3) IHXNUM

The heat exchanger sequence number.

(type = integer, default = none, units = none)

(4) IHXTYP

The type of the heat exchanger, counterflow or parallel flow.

(a) COUNTERFLOW
(b) PARALLEL

(type = character*16, default = none, units = none)

(5) HXAREA

Area used for heat exchanger.

(type = real, default = none, units = m2)

(6) HXHTC

Constant heat transfer coefficient.

(type = real, default = none, units = W/m2-K)

Or

Name of CF returning heat transfer coefficient.

(type = character*16, default = none, units = none)

(7) HXFL1

Name of first flow path for heat exchanger.

(type = character*16, default = none, units = none)

(8) HXFL2

Name of second flow path for heat exchanger.

(type = character*16, default = none, units = none)

Example

```
FL_IHX   1
         1    HX1 10 COUNTERFLOW 15.0 1.0e3 FL10 FL20
```

FL_CCF – Counter-Current Stratified Flow Model

Optional

This model simulates coupling by exchange of momentum between flows in two flow paths.

(1) NUMCCF
The number of pairs of coupled paths
(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NUMCCF; the first 7 fields are required. These may be followed by optional fields.

(1) NX
Table row index.
(type = integer, default = none, units = none)

(2) CCFNAME
The name of the counter-current flow model.
(type = character, default = none, units = none)

(3) CCFNUM
The counter-current flow model sequence number.
(type = integer, default = none, units = none)

(4) CCFFL1
Name of first flow path in coupled pair.
(type = character, default = none, units = none)

(5) CCFFL2
Name of second flow path in coupled pair.
(type = character, default = none, units = none)

(6) CHARLEN
Characteristic length for use in Froude number.
(type = real, default = none, units = m)

(7) C_D
Discharge coefficient to be used in Epstein-Kenton correlation.
(type = real, default = none, units = m)

The following optional fields may be used to directly specify the coefficients in the Froude correlation for volumetric flux. If one is specified, both must be. The defaults correspond to horizontal flow through a rectangular opening

C_FLOOD

Coefficient in Froude correlation for volumetric flux at flooding.

(type = real, default = 0.942809, units = none)

C_NC

Coefficient in Froude correlation for volumetric flux for pure natural convection (no net volumetric flow).

(type = real, default = 0.333333, units = none)

The following field is required if positive flow in one of the paths is reversed with respect to positive flow in the other.

DIRECT

Enter keyword 'REVERSED' to indicate such reversal.

(type = character, default = none, units = none)

If the two coupled flow paths do not connect the same pair of volumes, it is necessary to specify the control volumes from fluid densities for use in the Froude number will be obtained. If one is specified, both must be.

CCVOL1

Name of the first control volume providing fluid densities for use in the Froude number.

(type = character*16, default = none, units = none)

CCVOL2

Name of second control volume providing fluid densities for use in the Froude number.

(type = character*16, default = none, units = none)

Example

```
FL_CCF   3 ! Name   #   FL1     FL2    Len
1  'CCF1' 123 'Upper' 'Lower' 1.0
!                                          C_Flood C_NC
2  'CCF2' 4   'Path1' 'Path2' 0.5  0.9549  0.3038
!                                          CCVOL1  CCVOL2
3  'HL'   4   'Top'   'Bot'   0.8  UP_PLEN SG_IN
```

2.2 MELCOR User Input

Records FL_LME, FL_JLF, and FL_JLT may be included in MELCOR input.

Record FL_LME defines the length over which momentum exchange occurs between pool and atmosphere flows sharing a flow path. The value used affects entrainment and flooding by controlling the balance between momentum exchange (which tends to couple the two flows and reduce slip) and buoyancy (which tends to separate pool from atmosphere).

Records FL_JLF and FL_JLT define the junction openings in the "*From*" and "*To*" volumes, respectively, i.e., the range of elevations within those volumes "seen" by the flow path for purposes of allowing pool and/or atmosphere to flow. In some calculations, use of a restricted opening height may be necessary to prevent the flow of inappropriate phases during relatively quiescent segments of a calculation when gravitational separation should occur. On the other hand, extremely slow running of MELCOR might occur during relatively dynamic segments when entrainment would be expected (see discussion in Sections 1.1.2 and 1.1.3.). Also, the consequences of an initial choice may not become apparent until late in a long calculation. Therefore, MELCOR allows redefinition of the junction opening on restart of a calculation. Although the default definition in MELGEN of the momentum exchange length for a vertical flow path involves the junction opening heights, inclusion of these records in MELCOR does not lead to automatic redefinition of that length.

In addition to these limited modifications to the flow path models and database permitted in MELCOR input, several characteristics of control functions—including bounds on their values—may be modified on restart. This allows limited access to the definitions of pumps and valves. It may be used, for example, to select among a number of scenarios involving different failure paths, provided that all required paths were provided in the original MELGEN input.

3 Sensitivity Coefficients

Sensitivity coefficients for thermal hydraulic models are described in the CVH Package Users' Guide.

4 Plot Variables

The plot variables currently included in the flow path package are listed below, along with a brief description. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

FL-EFLOW.x.n	Enthalpy flow rate of pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path index n, with respect to the normal MELCOR reference point. (units = W)
FL-FRUNBLK.n	Fraction of flow path index n that is unblocked by core debris. This variable is available only for flow paths where the flow blockage model has been invoked by inclusion of a FL_BLK record as part of MELGEN input; for flow paths whose area is controlled by an ordinary valve,

	the open fraction may be plotted as the value of the related control function, CF-VALU.n. (units = dimensionless)
FL-I-EFLOW.x.n	Integral of enthalpy associated with pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path index n, with respect to the normal MELCOR reference point. (units = J)
FL-I-H2O-MFLOW.n	Integral of mass of water (*pool* + *fog* + *vapor*) flowing through flow path index n. (units = kg)
FL-I-MFLOW.m.n	Integral of mass of material m flowing through flow path index n. (units = kg)
FL-MFLOW.n	Total mass flow rate (all hydrodynamic materials) through flow path index n. (units = kg/s)
FL-MFLOW.m.n	Mass flow rate of material m through flow path index n. (units = kg/s)
FL-V-N-OC.n	Number of times valve index n has opened or closed. Any area fraction greater than 0.0 is considered "open." (units = dimensionless)
FL-VELLIQ.n	Velocity of pool through flow path index n. (units = m/s)
FL-VELVAP.n	Velocity of atmosphere through flow path index n. (units = m/s)
FL-VOID.n	Void fraction in flow path index n. (units = dimensionless)

<u>MCH Plot Variables</u>

FL- MCH-DELTA_P.n	Pressure boost in mechanical model index n. (units = Pa)

FL- MCH-EFFIC.n	Efficiency in mechanical model index n. (units = none)
FL- MCH- SPEED.n	Speed in mechanical model index n. (units = none)
FL- MCH- TORQUE.n	Torque in mechanical model index n. (units = Nm)
FL- MCH- POWER.n	Power in mechanical model index n. (units = W)

IHX Plot Variables

FL- IHX-TIN1.n	Inlet temperature in flow path 1 for heat exchanger index n. (units = K)
FL- IHX-TOUT1.n	Outlet temperature in flow path 1 for heat exchanger index n. (units = K)
FL- IHX-TIN2.n	Inlet temperature in flow path 2 for heat exchanger index n. (units = K)
FL- IHX-TOUT2.n	Outlet temperature in flow path 2 for heat exchanger index n. (units = K)
FL- IHX-HTCOEFF.n	Heat transfer coefficient for heat exchanger index n. (units = W/m^2-K)
FL- IHX-Q12.n	Heat transfer between flow paths for heat exchanger index n. (units = W)

MACCS Plot Variables

MACCS-RHONOM	Nominal aerosol density used to relate diameter and mass. (units = kg/m^3)
MACCS-nn-PLTEMP	Fluid temperature associated with release path index nn.

	(units = K)
MACCS-nn-PLHEAT	Enthalpy associated with release path index nn. (units = K)
MACCS-nn-PLMFLO	Cumulative fluid mass flow associated with release path nn. (units = kg)
MACCS-nn-PLMWT	Fluid molecular associated with release path index nn. (units = kg/mole)
MACCS-nn-M-RE-cc.ii	Released radioactive mass for RN class cc in size group (section) ii, associated with release path index nn. Vapor is in size group 0. Here, "cc" is the two-digit class number.

5 Control Function Arguments

The control function arguments currently included in the flow path package are listed below, along with a brief description. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

The choices permitted for NameMat always include 'POOL', 'FOG', 'H2O-VAP', or those other materials identified by input to the NonCondensible Gas (NCG) package. In certain cases (see below) the keywords 'TOTAL', 'WATER' are also interpreted to mean, respectively, the total contribution from all materials, or the total contribution from all water phases ('POOL', 'FOG', 'H2O-VAP').

FL-EFLOW(NameFP,x)	Enthalpy flow rate of pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path NameFP, with respect to the normal MELCOR reference point. (units = W)
FL-EFLOW-TR (NameFP,x)	Enthalpy flow rate of pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path NameFP, with respect to the normal MELCOR reference point, including any transformation within the path by SPARC or the flashing model. (units = W)

FL-FRUNBLK (NameFP)	Fraction of flow path that is unblocked by core debris. This variable is available only for flow paths where the flow blockage model has been invoked by inclusion of a FL_BLKFLnnnBk record as part of MELGEN input; for flow paths whose area is controlled by an ordinary valve, the open fraction may be plotted as the value of the related control function, CF-VALU(NameCF). (units = dimensionless)
FL-I-EFLOW (NameFP,x)	Integral of enthalpy associated with pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path NameFP, with respect to the normal MELCOR reference point. (units = J)
FL-I-EFLOW-TR(NameFP,x)	Integral of enthalpy associated with pool (*x* = 'P') or atmosphere (*x* = 'A') flow through flow path NameFP, with respect to the normal MELCOR reference point, including any transformation within the path by SPARC or the flashing model. (units = J) (units = kg)
FL-I-MFLOW (NameFP,NameMat)	Integral of mass NameMat flowing through flow path NameFP. The choices allowed for NameMat are '*WATER*' for water (*pool* + *fog* + *vapor*), '*POOL*', '*FOG*', '*H2O-VAP*', or those other materials identified by input to the NonCondensible Gas (NCG) package. (units = kg)
FL-I-MFLOW-TR(NameFP,NameMat)	Integral of mass of material NameMat flowing through flow path NameFP, including any transformation within the path by SPARC or the flashing model. (units = kg)
FL-MFLOW-TR(NameFP,NameMat)	Mass flow rate of material NameMat through flow path NameFP, point, including any transformation within the path by SPARC or the flashing model. (units = kg/s)

FL-MFLOW(NameFP,NameMat)	Mass flow rate of material NameMat through flow path NameFP or Total mass flow rate (all hydrodynamic materials) if *NameMat* = 'ALL'. (units = kg/s)
FL-V-N-OC (NameValve)	Number of times valve NameValve has opened or closed. Any area fraction greater than 0.0 is considered "open". (units = dimensionless) (units = m/s)
FL-VEL(NameFP,x)	Velocity of pool (*x* = 'P') or atmosphere (*x* = 'A') through flow path NameFP. (units = m/s)
FL-VOID (NameFP)	Void fraction in flow path NameFP. (units = dimensionless)

<u>MCH Control Function Arguments</u>

FL-MCH-DELTA_P(nameMCH)	Pressure boost in mechanical model nameMCH. (units = Pa)
FL-MCH-EFFIC(nameMCH)	Efficiency in mechanical model nameMCH. (units = none)
FL-MCH-SPEED(nameMCH)	Speed in mechanical model nameMCH. (units = none)
FL-MCH-TORQUE(nameMCH)	Torque in mechanical model nameMCH. (units = Nm)
FL-MCH-POWER(nameMCH)	Power in mechanical model nameMCH. (units = W)

<u>IHX Control Function Arguments</u>

FL-IHX-FL1-Tin(nameIHX)	Inlet temperature in flow path 1 for heat exchanger nameIHX. (units = K)
FL-IHX-FL1-Tout(nameIHX)	Outlet temperature in flow path 1 for heat exchanger nameIHX. (units = K)

FL-IHX-FL2-Tin(nameIHX)	Inlet temperature in flow path 2 for heat exchanger nameIHX. (units = K)
FL-IHX-FL2-Tout(nameIHX)	Outlet temperature in flow path 2 for heat exchanger nameIHX. (units = K)
FL-IHX-HTCoeff(nameIHX)	Heat transfer coefficient for heat exchanger nameIHX. (units = W/m^2-K)
FL-IHX-Q12(nameIHX)	Heat transfer between flow paths for heat exchanger nameIHX. (units = W)

<u>MACCS Control Function Arguments</u>

MACCS-RHONOM	Nominal aerosol density used to relate diameter and mass. (units = kg/m^3)
MACCS-PLTEMP(NameFP)	Fluid temperature associated with release path NameFP. (units = K)
MACCS-PLHEAT(NameFP)	Enthalpy associated with release path NameFP. (units = J)
MACCS-PLMFLO(NameFP)	Cumulative fluid mass flow associated with release path NameFP. (units = kg)
MACCS-PLMWT(NameFP)	Fluid molecular with associated with release path NameFP. (units = kg/mole)
MACCS-M-RE(NameFP,NameCLS,ii)	Released radioactive mass for RN class NameCLS in size group (section) ii associated with release path NameFP. Vapor is in size group 0. (units = kg)

MACCS-M-RE(NameFP,NameCLS,AER)
: Released radioactive aerosol mass for RN class NameCLS.
(units = kg)

MACCS-M-RE(NameFP,NameCLS,TOT)
: Total released radioactive mass for RN class NameCLS, including aerosol and vapor.
(units = kg)

6 Example Input

This section gives several examples of input to the FL package. Anything followed by an exclamation mark is a comment.

In the first example below, flow path named '*Door*' defines a door between control volumes CV10 and CV20; positive flow is from volume CV10 to volume CV20, as a simple opening that is 1m wide and 2m high. The elevation of the floor is 0m. Both junction altitudes are taken at the center of the door (1m); both junction opening heights are taken as 2m to include the full height of the door. The inertial length is estimated as 1m, representing a length of a few times the thickness of the wall; the segment length is an estimate of the wall thickness, and the hydraulic diameter is 4 times the area ($2m^2$) divided by the wetted perimeter (6m). Form loss is probably more important than wall friction; the loss coefficient is estimated to be 2.0 in both directions.

```
FL_ID   Door
FL_FT   CV10   CV20   1.0   1.0       ! Center alt = 1 m
! A L Open H-fm H-to Junction openings
FL_GEO   2.0   1.0   1.0   2.0    2.0 ! see full 2 m height
FL_JSW   3                             ! Normal, horiz
FL_USL   2.0   2.0             ! Forward and reverse loss coeffs
! Segment L, A, and hyd diam
FL_SEG   1 !NSEG SAREA SLEN  SHYD
             1    2.0   0.2  1.33
```

In the second example, flow path named '*Core to UP*' represents the vertical connection of the core (control volume CV120) to the upper plenum (control volume CV150) in a BWR; the direction of positive flow is from core to upper plenum. The elevation of the plane separating them is 10m. The form loss coefficients are 9.5 for forward flow and 10.5 for reverse flow. From the center of the core to the center of the upper plenum, fluid must travel 2m (half the height of the core) through a flow area of $4m^2$ and a hydraulic diameter of 0.01m, and then travel 1.6m through the more open upper plenum with an area of $8m^2$ and a hydraulic diameter of 0.8m. The nominal flow area is taken as $4m^2$. The inertial length was calculated from $L/ 4m^2 = 2m / 4m^2 + 1.6m / 8m^2$, which yields L = 2.8m; the precise value used is not critical.

```
FL_ID   'Core to UP'
FL_FT   CV120   CV150   10.0   10.0 ! CV120 to CV150
FL_GEO   8.0   2.8   1.0            ! A, L, Open fraction
```

```
FL_JSW   0   NoBubbleRise   NoBubbleRise ! Vert, no SPARC + or -
FL_USL   9.5   10.5 ! Forward, Reverse form losses
FL_SEG 2 !NSEG SAREA SLEN  SHYD
           1    4.0   2.0  0.01 ! Segment for half Core
           2    8.0   1.6  0.8  ! Segment for half UP
```

Because no junction opening heights were included on the FL_JSW record, the radius of a circle with area $4m^2$ will be used in each volume. If desired, the height of the junction openings could be reduced by including the desired values on the FL_GEO record. This could also be done by specifying the desired limits directly on FL_JLF and FL_JLT records; for example inclusion of the following records

```
FL_JLF   9.5   10.0 ! 'From' opening, 9.5-10.0 m
FL_JLT   10.0  11.0 ! 'To' opening, 10.0-11.0 m
```

would explicitly define the junction openings to see the range of elevations from 9.5 to 10.0m for forward flow out of volume CV120 and 10.0 to 11.0 for reverse flow out of volume CV150.

The default momentum exchange length for this flow path might limit countercurrent flow of pool from the upper plenum (volume CV150) and atmosphere from the core (volume CV120) sufficiently to result in levitation of water in the upper plenum under conditions of vigorous boiling in the core. Inclusion of a record such as

```
FL_LME   0.2 ! Momentum exchange length
```

would substantially increase the limits of countercurrent flow and reduce the levitation of water in the upper plenum.

Addition of the record

```
FL_VEL   0.0   4.0 ! Initial atmos, pool velocity
```

will define an initial pool velocity of 4m/s from core to upper plenum in the flow path.

Jet pumps in a BWR are frequently modeled as pipe-like flow paths.

The partial input

```
FL_ID 'Jet Pump'
FL_FT   CV110   CV100   9.0   4.0   ! From DC to LP
FL_GEO   0.63   5.0   1.0           ! A, L, Open fraction
...
FL_SEG   1 !NSEG SAREA SLEN  SHYD SRGH
             1   0.63   5.0  0.2  5.E-6 ! Dhyd for one of 20,
                                        ! smooth
```

represents 20 jet pumps in parallel, each of 0.2m diameter, with a total area of $0.63m^2$, connecting the downcomer (control volume CV110) at 9m elevation to the lower plenum (control volume CV100) at 4m elevation. Surface roughness is 5 µm rather than the default of 50 µm. The volume of the pumps is usually included in the lower plenum.

Increased flow resistance and blockage of flow paths by core debris may be modeled by use of FL_BLK records. For example, if flow path named 'LP to CHNL' connects the lower plenum to the channel region of the core of a BWR, use of the input

```
FL_ID   'LP to CHNL'
FL_FT   CV110   CV120   4.0   4.0   ! CV110 to CV120
...
FL_BLK   AXIAL-C   1   3   4   6 ! Channel flow blockage by COR
```

will include the resistance of debris in the channel in rings 1 through 3 of axial levels 4 through 6 of the core in computing flows through that flow path. The input will be rejected, and MELGEN will not generate an initial restart, unless the channel regions of all core cells involved lie in one of the control volumes (CV110 or CV120) connected by the flow path.

Similar input allows modeling of the opening of a flow path between the channel and bypass regions of a core. If flow path named 'CH to BP' connects the channel and bypass regions, use of the input

```
FL_ID   'CH to BP'
FL_FT   CV120   CV130   6.0   6.0 ! CV120 to CV130
...
FL_BLK   CHANNEL-BOX   1   3   4   6 ! Channel box failure
```

will base the open area and flow resistance of flow path 'CH to BP' on the state of the channel box in rings 1 through 3 of axial levels 4 through 6 of the core. The input will be rejected, and MELGEN will not generate an initial restart, unless the channel and bypass regions are distinguished in all core cells involved. Furthermore, the channel regions of all the cells must lie in one of the control volumes (CV120 or CV130) connected by the flow path and the bypass regions of all the cells in the other.

The partial input

```
FL_ID 'Pipe'
FL_FT   CV123   CV456   20.0   -10.0
FL_GEO   0.002   30.0   1.0   ! A, L, Open fraction
...
! A L Dhyd
FL_SEG   1 !NSEG SAREA SLEN  SHYD
             1   .002  0.5   0.05
             2   0.05  29.5  0.25
```

represents a vertical pipe, 30m long, connecting control volume CV123 at 20m elevation to control volume CV456 at –10m elevation. A short segment has an area of 0.002m^2, corresponding to a diameter of 50mm; the remainder has an area of 0.05m^2, corresponding to a diameter of 0.25m.

Addition of the record

```
FL_VLV  ! NV VLVNAME FLNAME KEYTRIP NVFONF  NVFONR   NVTRIP
          1   Valve1 'Pipe'  NoTRIP  CF20 ! no trip, open
                                         ! fraction from CF20
```

includes a simple (untripped) valve in the flow path. The fraction of its area that is open will be obtained from the value of control function CF20; a value of 1.0 corresponds to opening to the full 0.002m^2 nominal area.

If the flow path '*Pipe*' contains a check valve which restricts flow through the flow path to (say) the forward direction, it is preferable to model it as a "one-way" flow path rather than by defining a valve. This reduces the number of control functions that must be defined and avoids small reverse flows that can occur before the CF package can detect the reverse flow and close the valve. (This is a general consequence of the numerically explicit nature of all MELCOR control function models.) The required input record is

```
FL_JSW   10   ! Vertical, one-way forward
```

In this case, no FL_VLV record is required (or permitted).

Alternatively, addition of the record

```
FL_PMP      1 ! N PUMP NAME   FL NAME   TYPE     CF NAME
                1    Pump1     'Pipe'  QUICK-CF   CF200
FL199P0 QUICK-CF 200 * Control function pump
```

includes a pump (momentum source) in the flow path. The pump boost in pressure will be obtained from the value of control function CF200. See Section 7.7 of the CF Package Users' Guide for partial input regarding appropriate use of control functions to represent a homologous pump model similar to that in TRAC and RELAP.

Finally, the partial input

```
! VOLUMES JUNCT.ELEV
! FM TO FM TO
FL_ID 'Break'
FL_FT   CV150   CV300   10.0   10.0
...
FL_FLSH  1 ! NFL FLNAME FLSHDIR SC/CF/CONST CFNAME/DIAM  IFOGOPT
             1   'Break' FORWARD CONST       30.0E-6
! Flashing model on for forward flow, with Sauter mean droplet
! diameter 30 microns
```

will turn the flashing model on for flow into the "*To*" volume, CV150.

7 FL Package Output

Each printed edit generated by the FL package in MELGEN and MELCOR contains a snapshot description of the state of all flow paths, with the output organized in tabular form. The integrated (cumulative) flows are included. Many column headings are abbreviated, but most are relatively clear—particularly when it is understood that "LIQ" refers to pool and "VAP" to atmosphere, and that "ALPHA" is the fraction of the open area of a flow path occupied by atmosphere, and that "P HEAD" refers to pumps. For each valve, the number of times its state has changed from fully closed to at least

partially open or back is recorded as "NUMBER ON OR OFF." (A complete cycle, closed to open to closed is counted as 2.)

The "FLOW PATH TIME INDEPENDENT EDIT" is generated by the FL package in MELGEN and as part of the first edit for each MELCOR run. This edit lists properties of all flow paths, with the input flags, pump data, and valve data interpreted into English. The segment data associated with each flow path are listed, as are the flow paths connected to each control volume.

Users are strongly advised to check the initial edit generated by MELGEN—as well as the contents of the diagnostic file (MEGDIA)—before proceeding with a calculation. This is particularly important if FL_BLK input records were included in the input to model flow blockage effects in the COR package. When this is done, some of the values input to FL will be redefined for consistency with the geometry of COR. Any such values changed will be flagged by an asterisk ("*") in the edit, with the notation.

* MARKS DATA MODIFIED BY BLOCKAGE OR CHANNEL BOX MODEL

8 Diagnostics and Error Messages

Diagnostic messages will be written by MELGEN to report errors or inconsistencies in input. Typical errors include errors in record format and failure to supply all required input records. Inconsistencies between input to different packages are also identified; each of the volumes connected by a flow path must be defined and must include the specified elevation of the flow path junction; any tabular functions or control functions referred to by pump, valve, or time dependent flow path input must be properly defined.

Errors may propagate, resulting in messages that are apparently unrelated to the actual input records. In these cases, it is often necessary to rerun MELGEN with the identified errors corrected; the other error messages should then be eliminated or clarified.

No restart file will be written until all errors identified during input processing have been corrected. This does <u>not</u>, of course, assure that the accepted input properly describes the physical system that the user intends to model. The analyst should always examine the initial edit produced by MELGEN before proceeding to run MELCOR. We have found that an incorrect definition of control logic for a pump and valve model will frequently not be discovered until the logic is exercised—typically after the investment of significant computer resources. In many cases, the entire calculation must be rerun. A short preliminary run to test the logic, perhaps with simplified input or artificially modified initial conditions is often a good investment.

All time advancement calculations are performed by the CVH package. See Section 7.1 of the CVH Package Users' Guide for a description of errors related to time advancement in MELCOR.

Appendix 1

MELCOR 2.0	PTFRead	Comments
FL-EFLOW(NameFP,x) NameFP – the flow path name. The choices allowed for x are: x = 'A' for atmosphere; x = 'P' for pool.	FL-EFLOW.x.n n – the flow path index. The choices allowed for x are: x = 'A' for atmosphere; x = 'P' for pool.	Enthalpy flow rate of pool (x = 'P') or atmosphere (x = 'A') flow through flow path NameFP (n), with respect to the normal MELCOR reference point.
FL-FRUNBLK(NameFP) NameFP – the flow path name.	FL-FRUNBLK.n n – the flow path index.	Fraction of flow path that is unblocked by core debris. This variable is available only for flow paths where the flow blockage model has been invoked by inclusion of a FL_BLK record as part of MELGEN input; for flow paths whose area is controlled by an ordinary valve, the open fraction may be plotted as the value of the related control function, CFVALU(NameCF).
FL-I-EFLOW (NameFP,x) NameFP – the flow path name. The choices allowed for x are: x = 'A' for atmosphere; x = 'P' for pool.	FL-I-EFLOW.x.n n – the flow path index. The choices allowed for x are: x = 'A' for atmosphere; x = 'P' for pool.	Integral of enthalpy associated with pool (x = 'P') or atmosphere (x = 'A') flow through flow path NameFP (n), with respect to the normal MELCOR reference point.
FL-I-MFLOW (NameFP,NameMat) NameFP – the flow path name; NameMat – the material name. The choices allowed are: 'WATER' - for water (pool + fog + vapor); NCG material.	FL-I-MFLOW.m.n FL-I-H2O-MFLOW.n n – the flow path index; m- the material number. FL-I-H2O-MFLOW.n FL-I-MFLOW.m.n	Integral of mass of material NameMat (m) flowing through flow path NameFP.
FL-MFLOW	FL-MFLOW.n	Mass flow rate of material

MELCOR 2.0	PTFRead	Comments
(NameFP,NameMat) NameFP – the flow path name; NameMat – the material name. The choices allowed are: ALL – for all materials; NCG material.	FL-MFLOW.m.n n – the flow path index; m- the material number. FL-MFLOW.n FL-MFLOW.m.n	NameMat (m) through flow path NameFP (n) or Total mass flow rate (all hydrodynamic materials) if NameMat = 'ALL'.
FL-V-N-OC (NameValve) NameValve – the valve name.	FL-V-N-OC.n n – the valve index.	Number of times valve NameValve (n) has opened or closed. Any area fraction greater than 0.0 is considered "open".
FL-VEL(NameFP,x) NameFP – the flow path name. The choices allowed for x are: x = 'A' for atmosphere; x = 'P' for pool.	FL-VELLIQ.n FL-VELVAP.n n – the flow path index; FL-VELVAP.n FL-VELLIQ.n	Velocity of pool (x = 'P') or atmosphere (x = 'A') through flow path NameFP (n).
FL-VOID (NameFP) NameFP – the flow path name.	FL-VOID. n n – the flow path index.	Void fraction in flow path NameFP (n).
MACCS-RHONOM	MACCS-RHONOM	Nominal aerosol density used to relate diameter and mass.
MACCS-PLTEMP(NameFP) NameFP – the flow path name.	MACCS-n-PLTEMP n – the flow path index.	Fluid temperature associated with release path NameFP (n).
MACCS-PLMFLO(NameFP) NameFP – the flow path name.	MACCS-n-PLMFLO n – the flow path index.	Cumulative fluid mass flow associated with release path NameFP (n).
MACCS-PLMWT(NameFP) NameFP – the flow path name.	MACCS-nn-PLMWT n – the flow path index.	Fluid molecular with associated with release path NameFP (n).
MACCS-M-RE (NameFP, NameCLS, ii) NameFP – the flow path name. NameCLS – RN class name ii - size group (section)	MACCS-n-M-RE-cc.ii n – the flow path index. cc - class number ii - size group (section)	Released radioactive mass for RN class NameCLS in size group (section) ii associated with release path NameFP. Vapor is in size group 0.

Heat Structure (HS) Package
Users' Guide

The MELCOR Heat Structure (HS) package calculates heat conduction within an intact, solid structure and energy transfer across its boundary surfaces into control volumes. The modeling capabilities of heat structures are general and can include pressure vessel internals and walls, containment structures and walls, fuel rods with nuclear or electrical heating, steam generator tubes, piping walls, etc.

This users' guide provides basic information needed to run the HS package with the rest of MELCOR, including a short discussion of the nodalization scheme and calculational framework of the package and a detailed explanation of the user input and package output for MELGEN and MELCOR. Required and optional input, sensitivity coefficients, control function arguments, plot variables, and error messages are all covered.

More detailed information on the models and numerical solutions employed by the HS package can be found in the HS Package Reference Manual.

Contents

LIST OF FIGURES **5**

LIST OF TABLES **5**

1 INTRODUCTION **6**

1.1 Changes since Version 1.8.6 12

2 INPUT REQUIREMENTS **12**

2.1 MELGEN User Input 12

2.1.1 MELGEN Input for Heat Structures 12

HS_DFT – Heat Structure Package Default Scheme Record 13
HS_ID – Heat Structure Name and Sequence Number 13
HS_DELETE – Ignore all records associated with HS_ID 14
HS_GD – General Heat Structure Data 14
HS_EOD – Heat Structure Elevation and Orientation Data 15
HS_MLT – Heat Structure Multiplicity 16
HS_SRC – Internal Power Source Data 16
HS_ND – Node Data 17
HS_LB – Left (Inside) Boundary Surface Data 20
HS_LBR – Left (Inside) Boundary Surface Radiation Data 24
HS_LBP – Type of Flow, Critical Pool Fractions for Left (Inside) Boundary Surface 25
HS_LBT – Critical Temperature Differences, Scaling Factors for Left (Inside) Boundary Surface of Heat Structure 26
HS_LBF – Left Boundary Fluid Temperature Option 27
HS_LBS – Additional Data for Left (Inside) Boundary Surface 28
HS_RB – Right (Outside) Boundary Surface Data 29
HS_RBR – Right (Outside) Boundary Surface Radiation Data 31
HS_RBP – Type of Flow, Critical Pool Fractions for Pool and Atmosphere for Right (Outside) Boundary Surface of Heat Structure 31
HS_RBT – Critical Temperature Differences, Scaling Factors for Right (Outside) Boundary Surface of Heat Structure 32
HS_RBF – Right (Outside) Boundary Fluid Temperature Option 32
HS_RBS – Additional Data for Right (Outside) Boundary Surface 32

2.1.2 MELGEN Input for Gas Sources 33

HS_DG – General Gas Source Data 33
HS_DGSRC – Gas Source Characterization Data 36
HS_DGICE – Ice Condenser Structure Data 37

2.1.3 MELGEN Input for the Film-Tracking Model 38

HS_FT – Film-Tracking Model On/Off Switch 39

HS_FTDRN – Data for Drainage Heat Structures 39
HS_FTLBF – Data for Fraction of the Total Drainage from the Left Surface of HS 40
HS_FTLBM – Data for External Mass Source Rate for Left Surface 40
HS_FTLBE – Data for External Specific Enthalpy Source for Left Surface 41
HS_FTRBF – Data for Fraction of the Total Drainage from the Right Surface of HS 42
HS_FTRBM – Data for External Mass Source Rate for Right Surface 42
HS_FTRBE – Data for External Specific Enthalpy Source for Right Surface 42

2.1.4 MELGEN Input for Structure-to-Structure Radiation 44

HS_RD – Structure-to-Structure Radiation 44

2.2 MELCOR User Input 46

2.2.1 MELCOR Input for Heat Structures 46

2.2.2 MELCOR Input for Gas Sources 46

2.2.3 MELCOR Film-Tracking Model Input 47

2.2.4 MELCOR Input Controlling Structure-to-Structure Radiation 47

3 SENSITIVITY COEFFICIENTS 47

3.1 HS Sensitivity Coefficients 48

4000 – Surface Tension 48
4051 – Steady-State Iteration Parameters 49
4052 – Steady-State Iteration Weight Parameters 49
4055 – Transient Iteration Parameters 50
4056 – Transient Iteration Weight Parameters 50
4060 – Atmosphere Natural and Forced Convection Ranges 51
4061 – **4063** – Atmosphere Laminar and Turbulent Natural Convection Ranges 51
4064 – **4066** – Atmosphere Laminar and Turbulent Forced Convection Ranges 52
4071 – Bounds on Critical Pool Fractions 53
4080 – Pool Natural and Forced Convection Ranges 54
4081 – **4083** – Pool Laminar and Turbulent Natural Convection Ranges 54
4084 – **4086** – Pool Laminar and Turbulent Forced Convection Ranges 55
4101 – **4112** – Atmosphere Natural Convection 56
4113 – **4124** – Atmosphere Forced Convection 58
4151 – **4162** – Pool Natural Convection 59
4163 – **4174** – Pool Forced Convection 60
4180 – Nucleate Boiling Heat Flux 61
4181 – Critical Heat Flux 62
4182 – Minimum Film Boiling Heat Flux 62
4183 – Film Boiling Heat Flux 63
4184 – Radiation to Pool Heat Flux (Plane Model) 63
4200 – Mass Transfer Flux Model Transition Parameter 64
4201 – Sherwood Number for Diffusion Mass Transfer 64

4202 – Film Flashing Heat Transfer Coefficient 64
4203 – CVH Steam Depletion Control Parameters 65
4205 – Stainless Steel Melting (Degassing) Parameters 66
4210 – **4215**, **4220** – **4222**, **4230** – **4232** – Film Heat Transfer Coefficient Correlations 66
4251 – Minimum and Maximum Liquid Film Thickness 71
4252 – Film Modeling Options 71
4253 – Film-Tracking Model Correlation Parameters 71
1245 – Downward-Facing Lower Head Heat Transfer Correlations 73
HS_SC – HS Sensitivity Coefficients Input Record 74

4 PLOT VARIABLES 75

5 CONTROL FUNCTION ARGUMENTS 80

6 EXAMPLE INPUT 84

6.1 Example MELGEN Input 85
6.1.1 Heat Structure Input 85
6.1.2 Gas Source Input 86
6.1.3 Noncondensible Gas (NCG) Input 87
6.1.4 Material Property (MP) Input 87
6.1.5 Tabular Function (TF) Input 87
6.2 Example MELGEN Input for Ice Condenser Model 88
6.3 Example MELGEN and MELCOR Input for Film-Tracking Model 89

7 EXAMPLE OUTPUT 91

7.1 Time-independent Data Output 91
7.2 Time-Dependent Data Output 95
7.3 Additional MELGEN Output 97
7.4 Additional MELCOR Output 97

8 DIAGNOSTICS AND ERROR MESSAGES 97

8.1 MELGEN input Errors 97
8.2 MELCOR Input Errors 98
8.3 Requests to Repeat a Computational Cycle 98

APPENDIX 1 99

List of Figures

Figure 1.1 Heat Structure in a Control Volume 7

List of Tables

Table 3.1 Default Values for Sensitivity Coefficients 4101 – 4124 57
Table 3.2 Default Values for Sensitivity Coefficients 4151 – 4174 59

1 Introduction

The Heat Structure (HS) package calculates heat conduction within an intact, solid structure and energy transfer across its boundary surfaces into control volumes. This document is the users' guide for the HS package. It provides information that is necessary to execute the HS package with other packages in the MELCOR Code.

This section describes the modeling of a heat structure in the MELCOR Code and provides a brief discussion of the calculation procedure used to obtain the temperature distribution for each heat structure and to calculate its interactions with structures in other packages.

A heat structure is an intact, solid structure represented by one-dimensional heat conduction with specified boundary conditions at each of its two boundary surfaces. The modeling capabilities of heat structures are general and can include pressure vessel internals and walls, containment structures and walls, fuel rods with nuclear or electrical heating, steam generator tubes, piping walls, etc. Special logic also permits the representation of cylindrical ice columns to model pressurized water reactors (PWR) ice condensers.

Figure 1.1 illustrates a heat structure in a control volume. The heat structure is inclined at some angle with respect to the vertical and is partially immersed in a pool. Although the geometry shown here is rectangular, a heat structure may have a rectangular, cylindrical, spherical, or hemispherical geometry.

The heat structure in Figure 1.1 is nodalized with N temperature nodes. The nodalization is specified by user input and may be nonuniform, i.e. the distance between temperature nodes need not be the same. Node 1 is the temperature node at the left boundary surface for a rectangular geometry or at the inside boundary surface for a cylindrical, spherical, or hemispherical geometry. Node N is the temperature node at the right boundary surface for a rectangular geometry or at the outside boundary surface for the other geometries.

The region between two adjacent temperature nodes is called a mesh interval. Each mesh interval may contain a different material. The name of the material in each mesh interval must be specified by user input. The Material Properties (MP) package provides thermal properties for each material through an interface with the HS package. See the MP Package Users' Guide for details.

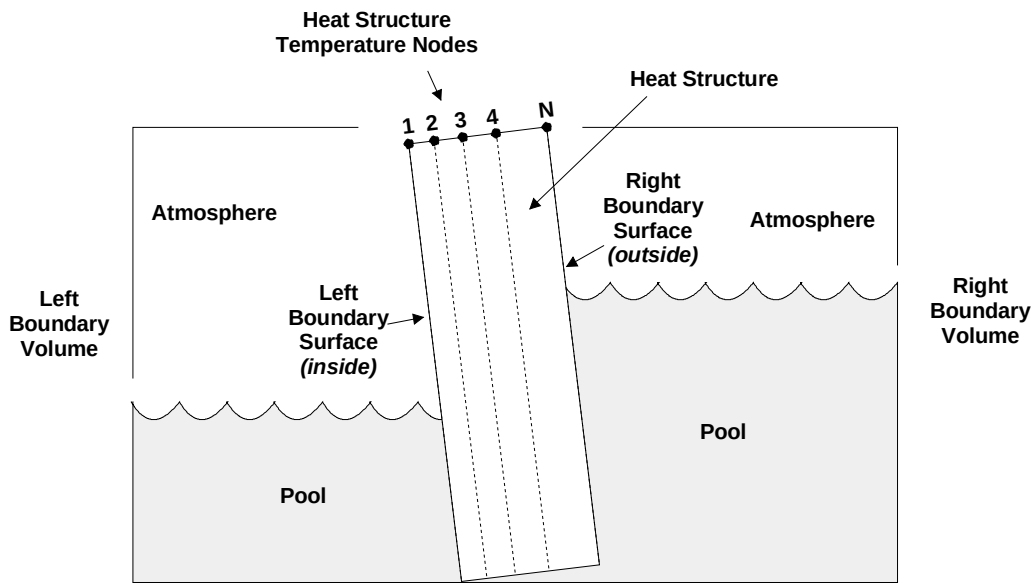


Figure 1.1 Heat Structure in a Control Volume

A pertinent issue for the user of the HS package concerns how finely a heat structure should be nodalized. While no definitive nodalization study is known to exist, the following considerations are useful in resolving this issue[1]. The mesh interval at a surface should be small enough to accurately respond to changes in the temperature of its boundary volume over a computational cycle. Accurate response is important for correctly calculating energy transfer, condensation-evaporation phenomena, and radionuclide deposition. However, the surface mesh interval should not be so small as to encourage temperature oscillations about the boundary volume temperature. Therefore, it is recommended that the size of the surface mesh interval be no smaller than the thermal diffusion length L_D. That is, ensure that

$$\Delta X_{surf} \geq L_D = \sqrt{4\, k\, \Delta t_{max} / C_p}$$

where

k	=	Thermal conductivity of material in surface mesh interval
Δt_{max}	=	Maximum computational timestep
C_p	=	Volumetric heat capacity (product of density and heat capacity) of material in surface mesh interval.

If this length is greater than the thickness of the heat structure in the direction of energy flow, two temperature nodes are recommended. Otherwise, to minimize the number of temperature nodes, the length of adjacent mesh intervals should increase toward the

[1] L.N. Kmetyk, MELCOR Assessment: Gedanken Problems, Volume 1, SAND92-0762, Sandia National Laboratories, Albuquerque, NM (January 1993).

interior of the heat structure. However, to maintain accuracy while increasing the distance between temperature nodes, it is recommended that the length of adjacent mesh intervals increase by no more than a factor of 2 within the same material, and that the length of adjacent mesh intervals be the same at the boundary of dissimilar materials.

Warning: If the length of the mesh interval at a surface is less than about half the thermal diffusion length, oscillations in the surface temperatures may occur. However, these oscillations are bounded and usually are damped out in a few computational cycles if the boundary conditions do not change much. To mitigate these oscillations, the current HS package utilizes a fully implicit numerical method rather than the Crank-Nicolson method used in earlier versions.

An internal power source may be specified for a heat structure. Its spatial dependence is specified by user input and may vary for each mesh interval. Its time dependence is given by a user-defined tabular function or control function.

Each heat structure has two boundary surfaces—left and right for rectangular geometries or inside and outside for cylindrical, spherical, or hemispherical geometries. At each boundary surface, one of the following boundary conditions is specified:

(1) Symmetry (adiabatic)

(2) Convective with calculated heat transfer coefficient

(3) Convective with calculated heat transfer coefficient and a specified surface power source function

(4) Convective with specified heat transfer coefficient function

(5) Specified surface temperature function

(6) Specified surface heat flux function.

If a convective boundary condition is selected for a boundary surface, a control volume must be specified as its boundary volume. No boundary volume is permitted for a symmetry boundary condition or a specified surface temperature boundary condition, and a boundary volume is a user option for the specified heat flux boundary condition. The Control Volume Hydrodynamics (CVH) Package Reference Manual provides information on control volume modeling.

If a boundary volume is specified for a surface, then some additional data are required through user input. For each boundary surface with a boundary volume, these data include

(1) Surface area

(2) Characteristic length (the dimension used in calculating the Reynolds, Grashof, Nusselt, and Sherwood numbers)

(3) Axial length (length of structure along boundary surface, used to determine pool fraction)

(4) Type of flow over the surface (internal or external; used in calculating the Nusselt number)

(5) Critical pool fractions for pool and atmosphere heat transfer.

The pool fraction of a heat structure boundary surface is the fraction of its surface area in the pool of its boundary volume. The critical value for the pool is the minimum value of the pool fraction for which heat transfer to the pool is calculated. The critical value for the atmosphere is the maximum value of the pool fraction for which heat transfer to the atmosphere and mass transfer at the surface are calculated.

The critical pool fractions provide the user with flexibility for calculating heat transfer at the surface of a heat structure. The user may elect to turn off the calculation of pool or atmosphere heat transfer by appropriate specification of these fractions. Turning off the calculation of pool heat transfer would be useful, for example, in a situation where a heat structure is partially immersed in a pool whose temperature is considerably less than the temperature of the atmosphere. The situation of high pool-atmosphere temperature differences could be calculated by the CVH package with its nonequilibrium thermodynamics option. In this situation, the heat transfer near the pool-atmosphere interface is significant, and axial conduction in this region of the heat structure is quite important. The HS package will calculate large energy transfers from the atmosphere to the pool under these conditions, since it calculates one-dimensional internal conduction and boundary conditions using a weighting of pool and atmosphere temperatures and heat transfer coefficients. If, during a calculation, large temperature differences between the pool and atmosphere are expected or axial conduction is expected to be important, the user should either subdivide a heat structure into structures of acceptable axial dimensions or turn off pool heat transfer by setting the critical pool fraction for pool heat transfer to 1.0.

If a convective boundary condition with calculated heat transfer coefficients is specified, then an extensive set of correlations is available for calculating natural or forced convection to the pool and atmosphere. Pool boiling heat transfer is calculated if the temperature of a heat structure surface is above the boundary volume saturation temperature by using correlations for nucleate boiling, critical heat flux, film boiling, and transition boiling.

Radiative heat transfer from a heat structure surface to the boundary volume pool is calculated during stable film and transition boiling. Radiative heat transfer can also be specified between a heat structure surface and the boundary volume atmosphere. Two options are currently available: an equivalent band model and a gray gas model.

Radiative heat transfer between user-defined pairs of heat structure surfaces may also be modeled.

Mass transfer between a heat structure surface and the boundary volume atmosphere is modeled using correlations or expressions for calculating mass flux. Models include condensation in a pure steam environment, condensation and evaporation in the presence of noncondensibles, and flashing in any environment. Liquid films on heat structure surfaces are also modeled so that condensate transferred from the boundary volume atmosphere and liquid deposited by other packages can be treated. Mass transfer affects the temperature distribution within a heat structure by its energy flux at the surface. When mass transfer occurs at a structure surface, an equation for the surface temperature of the resulting liquid film, including the energy flux due to mass transfer, is included in the set of conduction (temperature) equations for the structure.

A film-tracking model is available to treat film flow between structures. The user invokes this model by specifying, for each heat structure involved in this model, the list of heat structures to which this heat structure drains. The model uses a correlation for film thickness as a function of film flow and the conservation of mass to determine the film thickness and total rate of drainage from each structure in the network. The total drainage from each structure is then partitioned among three possible destination types specified by the user for each structure: (1) a user-defined fraction of the total drainage may be directed to one or more heat structure surfaces in the network, (2) a user-defined fraction may "rain" from the surface via a user-defined transfer process to the MELCOR Containment Spray (SPR) package, and (3) the remainder of the total drainage (one minus the sum of all the user-defined fractions listed above) is directed to the CVH pool component of the control volume associated with the structure surface. (This is the only destination for drainage from surfaces that are not included in film-tracking networks.) The model also allows the user to specify external water sources for any structure through tabular function or control function input. The model is primarily intended to treat the passive containment cooling systems proposed for advanced light water reactors (ALWRs), but it may also be used to track film flow in steam generators, etc.

Finite-difference equations are used to advance the temperature distribution of a heat structure in time during MELCOR execution, or to obtain its steady-state temperature distribution during MELGEN execution if specified by user input. These equations are obtained from an integral form of the one-dimensional heat conduction equation and boundary condition equations using a fully implicit numerical method. The finite-difference approximation is a tridiagonal system of N (or N + 1 or N + 2 if there is a liquid film on one or both surfaces of the structure) equations for a heat structure with N (or N + 1 or N + 2) temperature nodes. The solution of this system is obtained using the standard solution algorithm for a tridiagonal system of linear equations. The procedure is iterative because the coefficients are functions of temperature and must be updated between iterations.

A degassing model is provided for the release of gases from materials that are contained in heat structure mesh intervals. Input may be provided, for example, to

represent the release of water vapor or carbon dioxide from concrete as its temperature increases. The following information must be input by the user for each gas source in the degassing model:

(1) Heat structure surface from which the gas is released

(2) Number of mesh intervals over which the source is distributed

(3) Name of gas that is released.

Each gas source is characterized by the following information, which is provided by user input:

(1) Density

(2) Heat of reaction

(3) Boundaries of temperature range for the release of gas.

The mass of gas that is available for release is the source density times the volume of the mesh intervals over which the source is distributed. The HS package calculates a constant gas release rate over the degassing temperature range and modifies thermal properties over this range to account for the energy that is required for gas production and release. If the initial temperature of the structure exceeds the lower bound of the degassing temperature range, MELGEN issues a warning message to inform the user that the gas that would have been released below the initial temperature will be ignored and is unavailable for release.

A special application of the degassing model can be used to model ice condensers. In this application, the user defines a vertical cylinder with an adiabatic inner boundary and an associated gas source that releases liquid water into the pool associated with the outer radial boundary volume. The lower value of the degassing temperature range should be close to, but greater than, the freezing point of water (273.15 K) to avoid problems with the limits of the tabular functions in the MELCOR material properties libraries. The upper value of the degassing temperature range should be about 373 K to simulate sensible heating of the melted ice by counter-flowing steam. The heat of reaction should include the sensible heating of both the subcooled ice (from its actual temperature) and melted ice, in addition to the latent heat of fusion of water. The gas source density is the density of liquid water. Furthermore, the user is required to specify an initial temperature for the ice, and this temperature should coincide with the lower limit of the degassing range (274 K). The ice condenser modeling logic is activated by specifying TYPESRC = ICE in the user input for gas sources (record HS_DG).

This users' guide contains eight additional sections. Section 2 describes the user input required for utilizing this package in MELGEN and MELCOR execution. The sensitivity coefficients and their default values are given in Section 3. Lists and brief descriptions of the plot variables and control function arguments available in the HS package are found in Sections 4 and 5, respectively. Section 6 provides some example input that

illustrates the use of features offered in this package. Printed output is discussed in Section 7. Section 8 lists the diagnostics and error messages that are provided by this package.

1.1 Changes since Version 1.8.6

Two changes were made to the modeling of films on structure surfaces. These are described briefly below. A new sensitivity coefficient array, 4252, was added to allow these changes to be disabled, returning to the modeling used in MELCOR 1.8.5.

(1) When a rising pool covers an increasing fraction of a surface on which there is a film, MELCOR 1.8.5 did not transfer any of the film mass to the pool. In effect, the film mass was pushed up, ahead of the pool surface. By default, whenever the fraction of a surface covered by the pool increases, an appropriate fraction of the film mass is transferred to the pool.

(2) For surfaces not included in film-tracking networks, MELCOR 1.8.5 calculated a quasi-steady film thickness from a Nusselt-Rohsenow condensation model, and imposed it as a maximum, with any excess transferred to the pool. The model balances condensation rate and drainage rate, with the film thickness influencing both the condensation rate and the drainage rate. This neglect of drainage times is not justified in a transient environment and the calculation could become unstable. By default, the transient drainage calculation, previously used only for surfaces in film-tracking networks, is now used for <u>all</u> surfaces. In effect, each surface not explicitly named on the HS_FTDRN record is treated as a network of length 1.

2 Input Requirements

This section provides the input requirements for the MELCOR HS package, including a short description of the input quantities, their units, and their default values if any. See the HS Package Reference Manual for details.

2.1 MELGEN User Input

This section contains a detailed description of the MELGEN user input for the HS package. Specific input record descriptions are located in these sections:

Section 2.1.1 Heat Structure input records
Section 2.1.2 Gas Source input records
Section 2.1.3 Film-tracking input records
Section 2.1.4 Structure-to-Structure Radiation input records

2.1.1 MELGEN Input for Heat Structures

The following input records are required to specify a heat structure. A complete set of input records must be supplied for each heat structure. The first input record for the HS package must be HS_INPUT, followed by other records.

HS_DFT – Heat Structure Package Default Scheme Record

Optional

(1) DEFAULT

This record modifies one of the HS package sensitivity coefficient values to the 1.86 standard. If the record is present, one floating-point field must be present. Note that, if this record is present, it must follow HS_INPUT. This record can be specified once and is not needed for each heat structure (i.e., the record is not associated with any particular heat structure record accompanied by HS_ID).

(type = real, default = 2.0)

Examples

```
HS_DFT 1.86
```

If the DEFAULT field is 2.0, no change is made to the existing sensitivity coefficients. If the DEFAULT field is 1.86, the following HS package sensitivity coefficient will be modified, unless otherwise specified by the user input record.

Sensitivity Coefficients	DEFAULT = 1.86	DEFAULT = 2.0
4055 (2)	5.0E-04	0.5

HS_ID – Heat Structure Name and Sequence Number

Required

(1) HSNAME

User-defined heat structure name.
Single quotes should be used if the heat structure name contains blanks or if lower case characters are significant. This field may contain no more than 16 characters.

(type = character*16, default = none)

(2) IHSNUM

User-defined heat structure sequence number. Optional. If omitted, a sequence number will be assigned by incrementing the last sequence number by 10.

(type = integer, default = none)

Examples

```
HS_ID HeatSlab51501
HS_ID 'Heat Structure 51501'  51501
```

HS_DELETE – Ignore all records associated with HS_ID

Optional

This record is processed after all records in the input file are read. It will turn off all HS records related to the HSNAME identifier.

(1) HSNAME

User-defined heat structure name.

(type = character*16, default = none, units = none)

Example

```
HS_DELETE  HeatSlab51501
```

HS_GD – General Heat Structure Data

Required

(1) IGEOM

Indicator for type of geometry. No other values are permitted for this field.

(a) 1 or RECTANGULAR or RECT

(b) Rectangular geometry

(c) 2 or CYLINDRICAL or CYL
Cylindrical geometry.

(d) 3 or SPHERICAL or SPHERE
Spherical geometry.

(e) 4 or BOTTOMHALFSPHERE or B_HEMI
Hemispherical geometry of the bottom half of a sphere.

(f) 5 or TOPHALFSPHERE or U_HEMI
Hemispherical geometry of the top half of a sphere.

(type = integer / character*16, default = none, units = none)

(2) ISS

Steady-state initialization flag. The default is that a steady-state initialization will be performed.
Optional.

(a) 1 SS, or YES,
Steady-state initialization calculation is performed.

(b) 0 or NO

Steady-state initialization calculation is not performed and the initial temperature distribution is input on Record HS_ND.

(type = integer / character*3, default = 1, units = none)

Examples

```
HS_GD SPHERICAL NO
HS_GD 1         1
HS_GD 2         YES
```

HS_EOD – Heat Structure Elevation and Orientation Data

Required

(1) HSALT

Elevation parameter. This value is the elevation of the lowest point on the given heat structure. This parameter must satisfy the condition that each boundary surface of this heat structure is contained within its boundary volume.

(type = real, default = none, units = m)

(2) ALPHA

Orientation parameter. Rectangular geometries: the absolute value of ALPHA is the cosine of the acute angle α between a vertical line and the heat structure surface (see Figure 1.1).

(a) ALPHA = 1
Vertical surface.

(b) ALPHA = 0
Horizontal surface with left-hand side on the bottom.

(c) 0 < ALPHA < 1
Slanted surface with left-hand side on the bottom.

(d) ALPHA = -10^{-7}
Horizontal surface with the right-hand side on the bottom.

(e) $-1 < \text{ALPHA} < -10^{-6}$
Slanted surface with right-hand side on the bottom.

Cylindrical geometries: the absolute value of ALPHA is the cosine of the acute angle between a vertical line and the axis of this heat structure. Only the absolute value is used for cylindrical geometries.

Spherical or hemispherical geometries: ALPHA has no meaning and does not need to be entered.

For rectangular and cylindrical geometries, ALPHA must be greater than or equal to -1.0 and less than or equal to 1.0.

(type = real, default = none, units = none)

If only one field is present on this record, it will be interpreted as HSALT. In this situation, ALPHA will be set equal to 1.0, which corresponds to a vertically oriented heat structure.

NOTE on FLOORS and CEILINGS: In associating horizontal rectangular heat structures with CVH control volumes, the upper surface of the structure should be thought of as a floor for the volume. Therefore, for a heat structure of thickness DELT and ALPHA = 0, HSALT + DELT should be greater than or equal to the lowest elevation in the CVH volume-altitude table for that volume. In this case, the right-hand side of the structure is the floor of the CVH volume, and the CVH volume name is entered in the IBVR field of record HS_RB.

If ALPHA = -10^{-7} for the same heat structure, then the description is the same, except that the left- and right-hand sides are reversed. The right-hand side is on the bottom and the floor is the left-hand side. The CVH volume name is entered in the IBVL field of record HS_LB.

The same convention applies to ceilings of control volumes, which should be thought of as the lower surface of a horizontal heat structure.

Examples

```
HS_EOD 9.0   1.0
HS_EOD 13.815          1.0
```

HS_MLT – Heat Structure Multiplicity

Optional

(1) HSMULT

Multiplicity of heat structure. This value is the number of heat structures in this problem that are identical to heat structure HSNAME.

(type = real, default = 1.0, units = none)

Example

```
HS_MLT 1.
```

HS_SRC – Internal Power Source Data

Required

(1) ISRC

Internal power source type.

(a) -1 or CF

Power is given by a control function.

(b) 0 or NO

No internal power source exists for this heat structure.

(c) 1 or TF

Power is entered as a function of time in a tabular function.

(d) If ISRC is NO NAMEIPS and VSMULT are not needed.

If ISRC equals TF or CF, the power obtained from the TF or CF is assumed to be given in watts (W). In this case, QFRCIN on record HS_ND is required.

(type = integer / character*2, default = none, units = none)

(2) NAMEIPS

Name of CF or TF containing source power data.

This field is required if ISRC equals TF or 1; CF or -1.

(type = integer, default = none, units = none)

(3) VSMULT

Internal source multiplier.

This field is required if ISRC equals TF or 1 or CF or -1. It is the fraction of the power from the given tabular or control function that is applied to this heat structure.

(type = real, default = none, units = none)

Examples

```
HS_SRC NO
HS_SRC -1   CF500   1.0
HS_SRC TF   TF201   0.5
```

HS_ND – Node Data

Required

(1) NP

Number of temperature nodes. This value must be greater than 1.

(type = integer, default = none, units = none)

(2) NSTR

Number of lines in the following table. Optional field.

The user can input data for less than the actual number of temperature nodes if the nodalization between two or more nodes is equally spaced and the values of variables XI and TEMPIN change linearly and the values of MATNAM and QFRCIN are applied to all mesh intervals between these nodes. (See example below.)

(type = integer, default = NP, units = none)

The following data are input as a table with length NP:

(1) NN

Table row number.

(type = integer, default = none, units = none)

(2) N

Temperature node index that has location XI.

(type = real, default = none, units = none)

(3) XI

Location of temperature node N. XI must be greater than or equal to zero.

Rectangular geometries: XI is relative to any origin.

Cylindrical geometries: XI is relative to the axis of the cylinder.

Spherical geometries: XI is relative to the center of the sphere.

Hemispherical geometries: XI is relative to the center of the sphere that contains the hemisphere.

If XI for the first node is zero, then either the geometry of this heat structure must be rectangular (IGEOM = 1 on record HS_GD) or a symmetry boundary condition must be applied to the left boundary surface (IBCL = 0 on record HS_LB).

(type = real, default = none, units = m)

(4) TEMPIN

Initial temperature (must be positive). For a steady-state initialization, this value need not be provided and the placeholder '-' can be used.

(type = real/Character*8, default = none, units = K)

(5) MATNAM

Name of the material in the mesh interval with left-hand node N. This field may be a default material name or a new material name. If it is a new material name, then material property input must be provided. See the MP Package Users' Guide for a list of the default materials and the input required to specify a new material. MP

gases cannot be used here. Any gas must be input as pseudo-gas with user definition of the appropriate values for density, specific heat capacity, and thermal conductivity. The material name in the table line NSTR (last table line) should be omitted because the number of mesh intervals is equal to the number of temperature nodes minus one.

(type = character*24, default = none)

(6) QFRCIN

Relative source strength.

This record is required if ISRC in record HS_SRC equals CF or –1 or TF or 1.

The source values are relative and can be scaled by any factor. These values will be normalized during input processing and multiplied by the internal source multiplier (VSMULT on record HS_SRC) to provide the fraction of power from the appropriate tabular function that is applied to each mesh interval in heat structure HSNAME.

The relative strength in the table line NSTR (last table line) should be omitted because the number of mesh intervals is equal to the number of temperature nodes minus one.

(type = real, default = none, units = none)

Examples

```
HS_ND 5 !NN N XI          TEMPIN    MATNAM           QFRCIN
          1 1 0.65E-02     572.3    STAINLESS-STEEL  0.
          2 2 0.68E-02     568.7    STAINLESS-STEEL  0.2
          3 3 0.71E-02     565.3    STAINLESS-STEEL  0.3
          4 4 0.74E-02     562.0    STAINLESS-STEEL  0.5
          5 5 0.77E-02     558.8    !not needed
HS_ND 3 !NN N   XI        TEMPIN    MATNAM           QFRCIN
          1 1 0.0065      572.3     STAINLESS-STEEL
          2 2 0.0068      568.7     INSULATION
          3 3 0.0077      558.8
```

The following example illustrates two equal forms of input for this record; the full set of input data is recommended.

The full form of input data:

```
HS_ND 6 !NN N   XI         TEMPIN   MATNAM           QFRCIN
          1 1  0.65E-02    500.     STAINLESS-STEEL  0.2
          2 2  0.70E-02    600.     STAINLESS-STEEL  0.2
          3 3  0.75E-02    700.     STAINLESS-STEEL  0.2
          4 4  0.80E-02    700.     STAINLESS-STEEL  0.2
          5 5  0.90E-02    700.     CON              0.5
          6 6  1.00E-02    700.     !not needed
```

The value of TEMPIN is changed linearly and the intermediate nodes are equally spaced between the first and the fourth node locations, while MATNAM and QFRCIN are the same for these nodes. In this case, the short form of data input is acceptable:

```
HS_ND 6 4 !NN N  XI        TEMPIN   MATNAM           QFRCIN
           1 1  0.65E-02  500.     STAINLESS-STEEL  0.2
           2 4  0.80E-02  700.     STAINLESS-STEEL  0.2
           3 5  0.90E-02  700.     CON              0.5
           4 6  1.00E-02  700.     !not needed
```

HS_LB – Left (Inside) Boundary Surface Data

Required

(1) IBCL

Boundary condition type. The magnitude of IBCL must be one of the following integers to specify the type of boundary condition that is applied at the left (or inside) boundary surface of this heat structure.

(a) 0 or Symmetry
A symmetry (insulated) boundary condition is applied.

(b) 1 or CalcCoefHS
A convective boundary condition is applied with the heat transfer coefficients calculated by the HS package.

(c) 2 or Zukauskas
A convective boundary condition is applied with the heat transfer coefficients calculated by the HS package. The Zukauskas correlation for external flow across a tube bundle is used. The HS_ZUKL record is required.

(d) 10 or SourTimeTF
A convective boundary condition is applied with the heat transfer coefficients calculated by the HS package. A surface power source, whose power is given as a function of time by a tabular function, is also applied at this surface. The units of quantities obtained from the tabular function are assumed to be W.

(e) 20 or TempTimeTF
The temperature of this surface is given as a function of time by a tabular function. The units of quantities obtained from the tabular function are assumed to be K.

(f) 30 or FluxTimeTF
The heat flux from this surface is given as a function of time by a tabular function. The units of quantities obtained from the tabular function are assumed to be W/m^2.

(g) 40 or CoefTimeTF
A convective boundary condition is applied with the heat transfer coefficients given as a function of time by a tabular function. The units of quantities obtained from the tabular function are assumed to be W/m^2-K.

(h) 50 or CoefTempTF
A convective boundary condition is applied with the heat transfer coefficients given as a function of the surface temperature by a tabular function. The units of quantities obtained from the tabular function are assumed to be W/m^2-K.

(i) 60 = CoefCF
A convective boundary condition is applied with the heat transfer coefficients specified by a control function. The units of quantities obtained from the control function are assumed to be W/m^2-K.

(j) 70 or SourCF
A convective boundary condition is applied with the heat transfer coefficients calculated by the HS package. A surface power source, whose power is given by a control function, is also applied at this surface. The units of quantities obtained from the control function are assumed to be W.

(k) 80 or TempCF
The temperature of this surface is specified by a control function. The units of quantities obtained from the control function are assumed to be K.

(l) 90 or FluxCF
The heat flux from this surface is specified by a control function. The units of quantities obtained from the control function are assumed to be W/m^2.

Name of the tabular function or control function is input on field NAMECFTF below.

(type = integer / character*10, default = none, units = none)

(2) NAMECFTF

Name of a tabular function (for IBCL = 10, 20, 30, 40 and 50).
Name of a control function (for IBCL = 60, 70, 80 and 90)
This field is not input for IBCL = 0 and 1.

(type = character*16, default = none, units = none)

See the Tabular Function (TF) Package Users' Guide for details. If control functions are used to specify boundary conditions with IBCL = 60, 70, 80, or 90, their values must be initialized through their respective CF_ID records. See the Control Function (CF) Package Users' Guide for details.

(3) IBVL

Name of boundary volume. This value specifies the control volume that is associated with the left (inside) boundary surface of the given heat structure. If this value is NO, then no control volume is associated with this surface. Otherwise, all fluid quantities required for the heat transfer coefficient calculations are obtained from the appropriate quantities for the boundary volume specified by this value. It is noted that the control volume and heat structure elevations must be specified so that the heat structure clearly is located within the control volume.

(a) NO
If IBCL = 0, 20, or 80.

(b) Boundary Volume Name
If IBCL = 1, 10, 40, 50, 60 or 70.

(c) Any of the above values
If IBCL = 30 or 90,

(type = character*16, default = NO, units = none)

(4) MTEVAL

Mass transfer options:

(a) 0 or NO
Mass transfer at this surface is not evaluated. This is the only possible value if IBCL = 0, 20, 30, 80 or 90.

(b) 1 or YES
Mass transfer at this surface is evaluated. For IBCL = 1, 10, 40, 50, 60, and 70, both values of MTEVAL are permitted.

(type = character*3, default = NO, units = none)

(5) EnforceMax

This optional parameter is used to limit the film thickness to an arbitrary specified maximum value. The user can either specify this value for the maximum film thickness for this heat structure

(positive value) or provide a Boolean to indicate that the code should use the maximum film thickness specified for C4251(2). Note, however, that that the maximum film thickness (C4251(2)) is a global parameter used to determine the maximum film thickness for geometries when the Reynolds analogy cannot be made for mass transfer (see the reference manual for additional information).

(a) EnforceMAX (character string)
The film thickness will be limited by the maximum film thickness specified for C4251(2)

(b) Positive number (real)
The film thickness for this surface will be limited by this value.

(type = character*10 or real, default = film thickness is not limited by an arbitrary user maximum, units = none)

Control volume assignment (and, therefore, mass transfer) for adiabatic or specified temperature boundary conditions (IBCL = 0, 20, or 80) is prohibited. For these conditions, IBVL and MTEVAL do not need to be entered (the default will be NO) and records HS_LBP and HS_LBT below do not need to be entered. IBVL must be entered for any other boundary condition type.

Mass transfer to a surface with a specified heat flux boundary condition (IBCL = 30 or 90) is also prohibited, although the assignment of a control volume in this case is optional.

If no control volume is associated with this surface (IBVL = NO), then the boundary condition type must be adiabatic, specified temperature, or specified heat flux (IBCL = 0, 20, 30, 80, or 90). Furthermore, all fields on records HS_LBP and HS_LBT are ignored, and these records do not need to be entered.

If the heat flux at both surfaces is specified (IBCL = 0, 30, or 90 on record HS_LB and IBCR = 0, 30, or 90 on record HS_RB), then steady-state initialization must not be requested (ISS must be NO on record HS_GD) because a solution is either impossible or arbitrary.

All possible input for this record is summarized in the following table:

(1) IBCL		**(2) NAMECFTF**	**(3) IBVL**	**(4) MTEVAL**
integer	character			
0	Symmetry	(no input)	NO (default and may be omitted)	NO (default and may be omitted)
1	CalcCoefHS	(no input)	CVH name	NO / YES
2	Zukauskas	(no input)	CVH name	NO / YES
10	SourTimeTF	TF name	CVH name	NO / YES
20	TempTimeTF	TF name	NO (default and may be omitted)	NO (default and may be omitted)

(1) IBCL		(2) NAMECFTF	(3) IBVL	(4) MTEVAL
30	FluxTimeTF	TF name	NO / CVH name	NO (default and may be omitted)
40	CoefTimeTF	TF name	CVH name	NO / YES
50	CoefTempTF	TF name	CVH name	NO / YES
60	CoefCF	CF name	CVH name	NO / YES
70	SourCF	CF name	CVH name	NO / YES
80	TempCF	CF name	NO (default and may be omitted)	NO (default and may be omitted)
90	FluxCF	CF name	NO / CVH name	NO (default and may be omitted)

Examples

```
HS_LB Symmetry                    NO        NO
HS_LB 0
HS_LB FluxTimeTF  TF201           CVH-01
HS_LB 80          'CLEANING_ON/OFF'  NO        0
```

HS_LBR – Left (Inside) Boundary Surface Radiation Data

Optional

Inclusion of this record enables radiative heat transfer for the left surface.

(1) EMISWL

Wall emissivity of the left surface. A value of 0.0 turns radiative heat transfer off at this surface. The wall emissivity is constant for the transient, except when a water film is present. In this case, the emissivity of the wall is calculated using a mechanistic model for radiation on film-covered surfaces.

(type = real, default = none, units = none)

(2) RMODL

Radiation model employed. Upper and lower case characters are equivalent.

(a) 1 or EQUIV-BAND
Equivalent band model.

(b) 2 or GRAY-GAS-A
Gray gas model.

(type = integer / character*10, default = none)

(3) PATHL

Radiation path length for the left surface.

(type = real, default = none, units = m)

All three parameters must be entered if this record is present. If the record is not included in the input, no radiative heat transfer is calculated for this surface. See the HS Reference Manual for details.

Example

```
HS_LBR   1.0   GRAY-GAS-A   0.37E-01
```

HS_LBP – Type of Flow, Critical Pool Fractions for Left (Inside) Boundary Surface

Required

For boundary surfaces with specified or calculated heat transfer coefficients (IBCL = 1, 10, 40, 50, 60, or 70 on record HS_LB), fields IFLOWL and CPFPL are required, although IFLOWL is not used when heat transfer coefficients are specified as is the case for IBCL = 40, 50, or 60.

(1) IFLOWL

Indicator for type of flow over left (inside) boundary surface of the heat structure. This value is used to determine the type of convective heat transfer correlation to be used in evaluating the heat transfer coefficient on this surface Section 2.6.1 of the HS Reference Manual.

(a) -1 or INT
Internal flow.

(b) 1 or EXT
External flow.

No other values are permitted for this field.

(type = integer / character*3, default = none)

(2) CPFPL

Critical pool fraction. This is the minimum value of the pool fraction for which heat transfer to the pool is calculated at the left (inside) boundary surface of the heat structure. It must be between 0.0 and 1.0, inclusive. See the discussion below.

(type = real, default = none, units = none)

(3) CPFAL

Critical pool fraction for atmosphere. This is the maximum value of the pool fraction for which heat and mass transfer to the atmosphere are calculated at the left (inside) boundary surface of the heat structure. It must be between 0.0 and 1.0, inclusive. See the discussion below; also see sensitivity coefficient C4071 for further restrictions on the bounds.

Optional field.

(type = real, default = CPFPL, units = none)

Careful consideration should be given to the selection of values for CPFPL and CPFAL.

If CPFPL and CPFAL are set to the same value, heat transfer will occur to either the pool or atmosphere (but not both at the same time) depending upon the value of the pool fraction for the current time.

If CPFPL exceeds CPFAL, there will be no heat transfer to either pool or atmosphere while the value of the pool fraction is between CPFPL and CPFAL.

If CPFAL exceeds CPFPL, there will be heat transfer to both pool and atmosphere while the pool fraction is between CPFPL and CPFAL.

A non-fatal warning message is printed to the output and diagnostic files if the values of CPFPL and CPFAL are not equal.

Refer to the HS Reference Manual for further discussion of the implications of input values for CPFPL and CPFAL. Similar consideration should be given to the inputs for CPFPR and CPFAR, which apply to the right boundary surface on record HS_RBP.

Example

```
HS_LBP INT   0.05   0.95
```

HS_LBT – Critical Temperature Differences, Scaling Factors for Left (Inside) Boundary Surface of Heat Structure

Optional

(1) CTDPL

Critical temperature difference for pool heat transfer. If the temperature difference between the structure surface and the pool exceeds this value, then the timestep may be limited to ensure accuracy in the calculated temperature difference and stability in the pool temperature.

(type = real, default = 100., units = K)

(2) CTDAL

Critical temperature difference for atmosphere heat transfer. This value is used in exactly the same way as CTDPL, except it applies to heat transfer from the structure surface to the atmosphere rather than the pool.

(type = real, default = 100., units = K)

(3) XHTFCL

Calculated atmosphere heat transfer scaling factor. Calculated heat transfer coefficients (IBCL = 1, 10 or 70 on record HS_LB) to the atmosphere are multiplied by this arbitrary scaling factor, which must be nonnegative.

(type = real, default = 1.0, units = none)

(4) XMTFCL

Mass transfer scaling factor. Mass transfer coefficients (MTEVAL = YES and the control volume name IBVL is input on record HS_LB) for the condensation/evaporation of liquid films are multiplied by this arbitrary scaling factor, which must be nonnegative.

(type = real, default = 1.0, units = none)

Example

```
HS_LBT 100.0   100.0   1.0   1.0
```

HS_LBF – Left Boundary Fluid Temperature Option

Optional (but generally required for COR radial boundary heat structures; see the COR Package Users' Guide and the descriptions of input records COR_MS and COR_ZP).

This record provides the user with the option of choosing which bulk fluid temperature to use on each side in the heat transfer calculations for the specified heat structure.

(1) IOPTL

Options for left (inside) surface of the heat structure.

(a) -1 or CF

Use control function for boundary fluid temperature on the left-hand side.

(b) 0 or CV

Use bulk fluid temperature in left boundary CVH volume (either pool or atmosphere depending on liquid level) for boundary fluid temperature on the left-hand side.

(c) 1 or DTDZ

Use fluid temperature calculated by COR package dT/dz model for core cell for boundary fluid temperature on the left-hand side.

(type = integer, default = 0, units = none)

If IOPTL = CF then

(2) NAMECF

Name of the control function used for boundary fluid temperature on the left (inside) surface of the heat structure.

(type = character*16,default = none)

If IOPTL = DTDZ then

(2) NUMAXL

Number of axial level of COR cell adjusted to the given heat structure.

(3) NUMRAD

Number of radial ring of COR cell adjusted to the given heat structure.

(type = character*16,default = none)

Examples

```
HS_LBF CF      CF271
HS_LBF CV
HS_LBF DTDZ    1    2
```

HS_LBS – Additional Data for Left (Inside) Boundary Surface

Required if IBCL ≠ 0, 20, 80 or IBVL ≠ NO on record HS_LB.

This record is required for the left (inside) boundary surface of each specified heat structure, unless either (a) a symmetry or specified temperature boundary condition is applied at this surface (IBCL on record HS_LB is 0, 20, or 80) or (b) no boundary volume is specified for this surface (IBVL on record HS_LB is NO). In either case (a) or (b), these variables are not needed for the calculation.

(1) ASURFL

Area of boundary surface. This value is the area of the left (inside) boundary surface of the given heat structure. ASURFL must be strictly positive. This field is ignored for cylindrical or (hemi)spherical geometries.

Rectangular geometries. The area of this surface and the right (outside) boundary surface (ASURFR on record HS_RBS) must be equal.

Cylindrical geometries. The area of this surface will be calculated from the location of the left (inside) temperature node (XI on record HS_ND) and the axial length of this structure (BNDZL on this record).

Spherical or hemispherical geometries. The area of this surface will be calculated from the location of the left (inside) temperature node (XI on record HS_ND).

(type = real, default = none, units = m^2)

(2) CLNL

Characteristic length of boundary surface. This value is the dimension used to calculate quantities such as the Reynolds, Grashof, Nusselt, and Sherwood numbers. CLNL must be strictly positive.

(type = real, default = none, units = m)

(3) BNDZL

Axial length of boundary surface. This value is the dimension of the left (inside) boundary surface of the given heat structure in a direction perpendicular to the direction of energy flow within this heat structure. BNDZL must be strictly positive. This field is ignored for spherical or hemispherical geometries.

Rectangular or cylindrical geometries. The axial length of this surface and the right (outside) boundary surface (BNDZR on record HS_RBS) must be equal.

Spherical or hemispherical geometries. The axial length is calculated from the location of the left (inside) temperature node (XI on record HS_ND).

(type = real, default = none, units = m)

Example

```
HS_LBS 0.0786 0.11966 0.189
```

HS_ZUKL – Additional Data for Zukauskas heat transfer coefficient on left boundary

Required if IBCL = 2.

This record is required for the left (inside) boundary surface if the Zukauskas heat transfer coefficient is requested by the user (IBCL=2). See figure

(1) DIAM

Diameter of the tube

(type = real, default = none, units = m)

(2) ST

Transvers pitch for tube bundle. If '-' then default is accepted.

(type = real, default = 10.0 * DIAM, units = m)

(3) SL

Longitudinal pitch for tube bundle. If '-' then default is accepted.

(type = real, default = 10.0 * DIAM, units = m)

(4) Aligned or Staggered

User indicates whether tube bundle is aligned or staggered. If '-' then default is accepted.

(a) Aligned
Tube bundles are aligned.

(b) Staggered
Tube bundles are staggered

(type = = character*16, default = Aligned)

(6) CORRECTION

User indicates whether a correction factor is to be provided on the following field or whether the correction factor is to be determined from the number of rows specified on the next field.

$$Nu_D = MULT * CRe_{D,max}^{m} Pr^{n} \left(\frac{Pr}{Pr_s}\right)^{0.25} \quad MULTIPLIER, \qquad or$$

$$= C_2(N_L) CRe_{D,max}^{m} Pr^{n} \left(\frac{Pr}{Pr_s}\right)^{0.25} \quad ROWS$$

(a) MULTIPLIER (or MULT)
User provides a multiplier

(b) ROWS
User specifies number of rows

(type = = character*16, default = ROWS)

(6a) MULTIPLIER

Multiplier on correlation or if '-' or absent, then default is accepted.

(type = REAL, default = 1.0, units = none)

(6b) ROWS

Number of tube rows in calculation of correction factor or if '-' or absent, then default is accepted.

(type = integer, default = 20, units = none)

ROWS	1	2	3	4	5	7	10	13	16	20
Aligned	0.70	0.80	0.86	0.90	0.92	0.95	0.97	0.98	0.99	1.0
Staggered	0.64	0.76	0.84	0.89	0.92	0.95	0.97	0.98	0.99	1.0

Example

```
HS_LB Zukauskas 'SG'   YES
```

```
!            D        ST      SL     Aligned/Staggered      NROWS
HS_ZUKL    .015       .02     .02          Aligned          ROWS 20
```

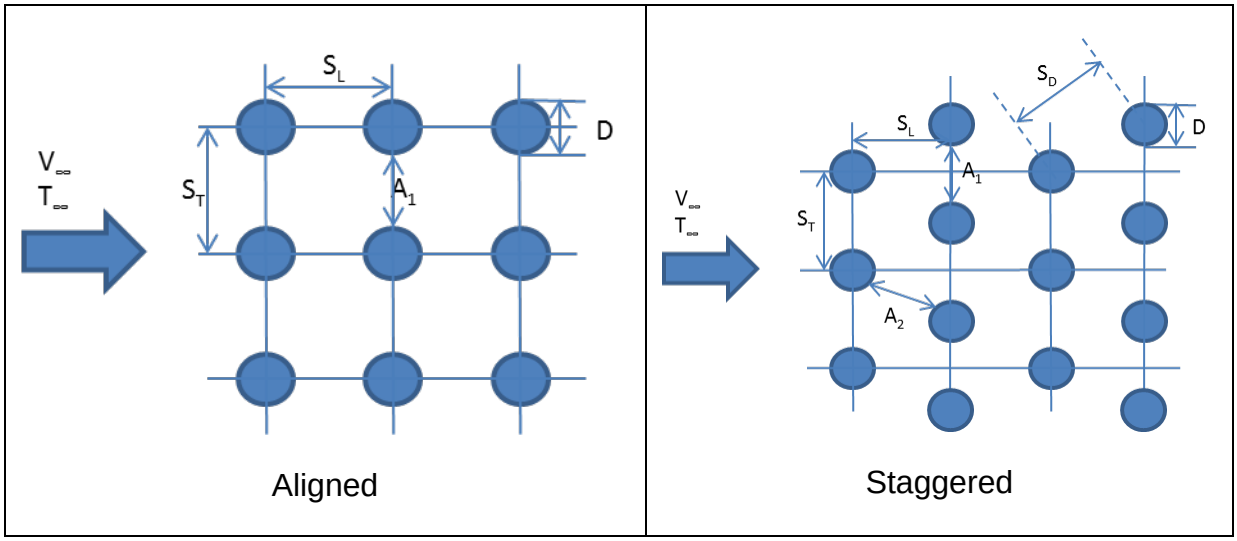


Figure 2. Tube bundle geometry for Zukauskas correlation.

HS_RB – Right (Outside) Boundary Surface Data

Required

The fields on this record (IBCR, NAMECFTF, IBVR, MTEVAR) correspond to the fields on record HS_LB, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface.

Example

```
HS_RB CalcCoefHS CV200            YES
```

HS_RBR – Right (Outside) Boundary Surface Radiation Data

Optional

The fields on this record (EMISWR, RMODR, and PATHR) correspond to the fields on record HS_LBR, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface.

Example

```
HS_RBR   1.0   GRAY-GAS-A   0.1
```

HS_RBP – Type of Flow, Critical Pool Fractions for Pool and Atmosphere for Right (Outside) Boundary Surface of Heat Structure

Required

The fields on this record (IFLOWR, CPFPR, CPFAR) correspond to the fields on record HS_LBP, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface.

Example

```
HS_RBP   INT   0.1    0.1
```

HS_RBT – Critical Temperature Differences, Scaling Factors for Right (Outside) Boundary Surface of Heat Structure

Optional

The fields on this record (CTDPR, CTDAR, XHTFCR and XMTFCR) correspond to the fields on record HS_LBT, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface.

Example:

```
HS_RBT   100.0   100.0   1.0   1.0
```

HS_RBF – Right (Outside) Boundary Fluid Temperature Option

Optional

The fields on this record (IOPTR and NAMECF) correspond to the fields on record HS_LBF, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface. But no core cell input is allowed for the right-hand surface; thus, the field IOPTR has only two options: CF and CV.

Example

```
HS_RBF   CF   'CLEANING_ON/OFF'
HS_RBF   CV
```

HS_RBS – Additional Data for Right (Outside) Boundary Surface

Required if IBCR ≠ 0; or IBVR ≠ NO.

This record is required for the right (outside) boundary surface of each specified heat structure, unless either (a) a symmetry boundary condition is applied at this surface (IBCR on record HS_RB is 0) or (b) no boundary volume is specified for this surface (IBVR on record HS_RB is NO). In either case (a) or (b) these variables are not needed for the calculation.

The fields on this record (ASURFR, CLNR, and BNDZR) correspond to the fields on record HS_LBS, but they apply to the right (outside) boundary surface of the given heat structure rather than the left (inside) boundary surface. However, in geometries for which the area of this surface is calculated, the location of the right (outside) temperature node is used, rather than the location of the left (inside) temperature node.

Example

```
HS_RBS 0.0845 0.15766 0.189
```

2.1.2 MELGEN Input for Gas Sources

The following input records are required to specify a gas source. A complete set of input records must be supplied for each gas source.

HS_DG – General Gas Source Data

Required

(1) NAMESRC

Gas source name.

(type = character*16,default = none)

(2) NAMEHS

Name of the heat structure that contains this source.

(type = character*16,default = none)

(3) TYPESRC

Indicator for the degassing model.

(a) 0 or GAS
Gas source.

(b) 1 or ICE
Ice condenser model is activated.

(c) 2 or STEEL
Steel model is activated.

(type = integer / character*5, default = 0, units = none)

(4) ISRCHS

Gas release surface. This value indicates from which heat structure boundary surface gas is released for gas source NAMESRC.

(a) -1 or LEFTBND
Gas is released from the left (inside) boundary surface of the heat structure.

(b) 1 or RIGHTBND

Gas is released from the right (outside) boundary surface.

(type = integer / character*8, default = none, units = none)

(5) ISDIST

Source distribution. This value is the number of mesh intervals over which the source NAMESRC is distributed. ISDIST must be greater than or equal to 1 and less than or equal to the number of mesh intervals in the heat structure (this equals NP-1, NP from record HS_ND) that contains this source.

(type = character*16,default = none)

(6) GASNAM

Name of the released gas. This is the name of the gas that is released from source NAMESRC. It must be the name of a material that is present in the problem. Valid names are

If TYPESRC = ICE

(a) POOL
Pool liquid water

If TYPESRC = GAS,

(a) FOG
Atmosphere liquid water

(b) H2O-VAP
Atmosphere water vapor

(c) H2
Hydrogen

(d) O2
Oxygen

(e) CO2
Carbon dioxide

(f) CO
Carbon monoxide

(g) N2
Nitrogen

(h) NO
Nitrogen monoxide

(i) N2O
Nitrous oxide

(j) NH3
Ammonia

(k) C2H2
Acetylene

(l) CH4
Methane

(m)C2H4
Ethylene

(n) GASNAM
User-defined gas

If TYPESRC = STEEL

(a) SS
Steel

The first three are present in all problems. The others must be specified for the problem through Noncondensible Gas (NCG) input. See the NCG Package Users' Guide for details on input for those gases that must be specified.

The SS material type is intended for use in MELCOR calculations where the COR package is employed to model melting of core boundary steel structures. Molten steel is entered into the outer radial ring of the core region at the corresponding elevation. Subsequent relocation of the steel is modeled by the COR package. If COR is not used, the SS source will be ignored.

(type = character*8, default = none)

The ice condenser model is activated by specifying TYPESRC = ICE. In this case, the heat structure should be cylindrical with an adiabatic inner boundary. The ice condenser model can be activated only for external flow (IFLOWR = EXT on the HS_RBP record) over the right boundary surface (ISRCHS = RIGHTBND on the HS_DG record). The associated "gas" source (which in this special case is actually water) that releases liquid water is connected to the CVH control volume specified by IBVR on record HS_RB. IBCR = 1 on record HS_RB, CPFAR = 0.5, and CPFPR = 0.5 on record HS_RBP should be used. In this case, record HS_DGICE must be input.

The initial temperature specified on record HS_ND should be 274 K. It is recommended that the value of HTRSRC on record HS_DGSRC be adjusted accordingly to include the sensible heat required to raise the temperature of the subcooled ice through the melting point to 274 K.

Caution: The ice condenser model will malfunction if any part of the ice condenser becomes submerged in the melt water pool in the outer boundary CVH volume. The user must provide sufficient volume below the bottom of the ice condenser structure to accommodate all of the melted ice. Also an initial ice temperature below 273.15 K will cause problems in the material properties routines.

There are several restrictions concerning the SS degassing option (TYPESRC = STEEL). These are listed below:

(1) The structure identified by NAMEHS on this record must have an orientation that is either horizontal (ALPHA = 0.0) or vertical (ALPHA = 1.0) as identified on the corresponding input record HS_EOD. If the structure is not horizontal or vertical, MELGEN processing will terminate.

(2) The structures for which SS degassing is desired must lie either along the core or above the core. If along the core, then a structure must align with one of the COR package axial segments as identified by input record COR_ZP. Structures are not permitted to span across COR axial segments. If the degassing structure lies above the core, then its lowermost elevation must exceed the uppermost core elevation as modeled by the COR package.

(3) Special restrictions apply to changing SS degassing input for MELCOR restarts. See Section 2.2.2.

Example

```
HS_DG SRC1    HeatSlab51501 ICE      RIGHTBND      5      POOL
HS_DG SRC2    HeatSlab_01   STEEL    LEFTBND       5      SS
HS_DG SRC3    HS2           GAS      LEFTBND       5      FOG
```

HS_DGSRC – Gas Source Characterization Data

Required

(1) RHOSRC

Density of gas source. This density times the volume of the mesh intervals over which it is distributed is the mass of gas that can be released by this source. RHOSRC must be greater than or equal to zero.

(type = real, default = none, units = kg/m^3)

(2) HTRSRC

Heat of reaction of gas source. The volumetric heat capacity is increased by the product of HTRSRC and the density of this source (RHOSRC on this record) divided by the temperature range over which gas is released (TEMPU minus TEMPL on this record) to yield the effective volumetric heat capacity of the material in a mesh interval of a degassing heat structure. HTRSRC must be greater than or equal to zero.

(type = real, default = none, units = J/kg)

(3) TEMPL

Lower temperature bound for degassing. This value is the lower boundary of the temperature range for the release of gas from gas source NAMESRC. TEMPL must be greater than zero.

(type = real, default = none, units = K)

(4) TEMPU

Upper temperature bound for degassing. This value is the upper boundary of the temperature range for the release of gas from gas source NAMESRC. TEMPU must be greater than TEMPL on this record.

(type = real, default = none, units = K)

Example

```
HS_DGSRC   100.0   0.5E+07   300.0   750.0
```

HS_DGICE – Ice Condenser Structure Data

Required if TYPESRC = ICE.

The next three variables apply only if an ice condenser structure has been defined by input TYPESRC = ICE on record HS_DG.

(1) HTCICE

Ice condenser Nusselt number multiplier. This value is a multiplier for calculated heat and mass transfer coefficients.

(type = real, default = none, units = none)

(2) RNDICE

Ice condenser RN deposition surface area enhancement factor. The surface area used in the RN deposition model is equal to the calculated ice surface area (defined below) plus the value of the original ice surface area multiplied by this enhancement factor. A nonzero value for RNDICE is used to account for heat transfer and radionuclide deposition to the wire baskets normally found in ice condensers that remain in place after the ice has melted.

(type = real, default = none, units = none)

(3) EXPICE

Ice condenser surface area exponent. The calculated heat and mass transfer surface area of the ice condenser is given by

$$A_o*[RNDICE + (1 - RNDICE)*(V/V_o)EXPICE]$$

where A_o is the original surface area of the ice (ASURFR on record HS_RBS), V_o is the original volume of the ice, and V is the current volume of the unmelted ice.

(type = real, default = none, units = none)

The values of HTCICE, RNDICE, and EXPICE should be adjusted to obtain the desired ice condenser performance. The value of EXPICE dictates how surface areas change with melting. A value of 1.0 implies that the area decreases linearly with the volume and is appropriate for transients that lead to rapid axial melting of the ice columns. A value of 0.5 implies a predominantly radial melt typical of slower transients.

Example

```
HS_DGICE   1.2  0.33  1.0
```

2.1.3 MELGEN Input for the Film-Tracking Model

The following input records are required to describe a network of heat structures connected for the purposes of film tracking. The user must define a set of HS_FT records for each heat structure in the network. The film-tracking model automatically calculates the total amount of water that drains off each heat structure surface in the network, and the user provides destination fractions to determine where that drainage goes. The destination fractions allow the user to divide the total drainage among any or all of three types of drainage destinations:

(1) The surface of one or more heat structures in the same network.

(2) The MELCOR SPR package (via a user-defined transfer process, see the TP Package Users' Guide) for treatment as "rain" from the surface (this is appropriate for inverted surfaces of which condensation is more likely to drip than drain, e.g. ceilings).

(3) The pool component of the CVH control volume associated with the given surface.

There may be more than one network in a problem (e.g., one network in the reactor vessel, one network in the steam generators, one network for containment structures, etc.), but no heat structure should be included in more than one network. Furthermore, recirculating drainage is not permitted; that is, heat structure B should not drain to heat structure A if heat structure A is included in the chain of heat structures that drain to heat structure B. All heat structures that receive film drainage from another heat structure or an external source should be included in the network (even though the bottom-most heat structures do not themselves drain to lower structures). A switch must be set ON on the corresponding HS_FT record for each heat structure film-tracking model. If the switch is on and a heat structure drains to one or more heat structures, then these heat structures must be listed on the HS_FTDRN record, and all HS_FT records are required. The switch for the bottom-most heat structure must also be set to ON. If the switch is OFF, then all HS_FT records must be absent. Heat structure surfaces that receive film drainage from other heat structures or external sources must support mass transfer; that is, the value of MTEVAL must be YES, and IBVL must be defined on record HS_LB and/or the same conditions must be defined for MTEVAR and

IBVR on record HS_RB if drainage is to be tracked on the left and/or right surface(s) of the given heat structure. Finally, the sum of drainage fractions (FRAINs [records HS_FTLBF/FTRBF] plus all FDRNs [record HS_FTDRN] for s = L or s = R) for each surface of a heat structure in the network must not exceed one. If the sum of drainage fractions for a surface is less than one, then the excess drainage will go to the pool of the CVH volume associated with that surface.

HS_FT – Film-Tracking Model On/Off Switch

Required

(1) IFTNUM

Film-tracking model on/off switch.

(a) 0 or OFF
(b) 1 or ON

(type = integer / character*3, default = 0, units = none)

Example

```
HS_FT OFF
```

HS_FTDRN – Data for Drainage Heat Structures

Required if IFTNUM = ON.

These records identify and describe the heat structures contained in the network.

(1) NUMDRN

Number of heat structures to which this structure drains. The value 0 means that this is the bottom-most heat structure in the network.

(type = integer, default = none, units = none)

The following data are input as a table with number of rows = NUMDRN:

(2) N

Table entry index

(type = integer, default = none, units = none)

(3) IDSDRN

Name of heat structure receiving the film drainage.

(type = character*16, default = none)

(4) FDRNL

The fraction of the total film drainage from the left surface of the given heat structure that goes to the left surface of heat structure IDSDRN.

(type = real, default = none, units = none)

(5) FDRNR

Same as FDRNL except it applies to the right surface of heat structures.

(type = real, default = none, units = none)

Example

```
HS_FTDRN 3 ! N IDSDRN         FDRNL       FDRNR
            1 HeatSlab51501   0.65E-02    0.573E-02
            2 HS2             0.68E-02    0.569E-02
            3 HeatSlab_01     0.77E-02    0.559E-02
HS_FTDRN 0
```

HS_FTLBF – Data for Fraction of the Total Drainage from the Left Surface of HS

Required if IFTNUM = ON.

(1) FRAINL

Fraction of the total drainage from the left surface of the heat structure that is transferred to the SPR package as "rain".

(type = real, default = none, units = none)

(2) NTPL

The *in* transfer process name for "rain" transfers to the SPR package from the left surface of the structure.
This field is required if FRAINL $\neq$ 0.

(type = character*16, default = none)

Example

```
HS_FTLBF 0.0
```

HS_FTLBM – Data for External Mass Source Rate for Left Surface

Optional, permitted if IFTNUM = ON.

This record is required if an external source of water is to be provided for the left surface of the heat structure.

(1) IMSRCL

Mass source rate (kg/s) identifier switch for left surface.

(a) -1 or CF
The source is obtained from a control function.

(b) 0 or NO
There is no source.

(c) 1 or TF
The source is obtained from a tabular function.

(type = integer / character*2, default = NO, units = none)

If IMSRCL ≠ NO then

(2) NAMECFTF
Name of the control function (for IMSRCL = CF).
Name of the tabular function (for IMSRCL = TF).

(type = integer / character*2, default = none, units = none)

Example

```
HS_FTLBM NO
```

HS_FTLBE – Data for External Specific Enthalpy Source for Left Surface

Optional, permitted if IFTNUM = ON.

This record is required if an external source of water is to be provided for the left surface of the heat structure.

(1) IESRCL
Specific enthalpy (J/kg) source identifier switch for source associated with IMSRCL.

(a) -1 or CF
Indicates a control function.

(b) 0 or NO
There is no source.

(c) 1 or TF
Indicates a tabular function.

(type = integer / character*2, default = none, units = none)

If IESRCL ≠ NO then

(2) NAMECFTF
Name of control function (for IESRCL = CF).
Name of tabular function (for IESRCL = TF).

(type = integer / character*2, default = none, units = none)

Examples

```
HS_FTLBE CF      CF205
HS_FTLBE 1       TabFun205
HS_FTLBE NO
```

HS_FTRBF – Data for Fraction of the Total Drainage from the Right Surface of HS

Required if IFTNUM = ON.

The fields on this record (FRAINR, NTPR) correspond to the fields on record HS_FTLBF, but they apply to the right surface of the given heat structure.

Example

```
HS_FTRBF 0.0
```

HS_FTRBM – Data for External Mass Source Rate for Right Surface

Optional, permitted if IFTNUM = ON.

The fields on this record (IMSRCR, NAMECFTF) correspond to the fields on record HS_FTLBM, but they apply to the right surface of the given heat structure.

Example

```
HS_FTRBM NO
```

HS_FTRBE – Data for External Specific Enthalpy Source for Right Surface

Optional, permitted if IFTNUM = ON.

The fields on this record (IESRCR, NAMECFTF) correspond to the fields on record HS_FTLBE but apply to the right surface of the given heat structure.

Example

```
HS_FTRBE NO
```

MELCOR 2.0 allows an alternate approach of a network definition using an INCLUDE directive. The switch for each heat structure is input on the HS_FT record in the original input file. All data concerning the film-tracking model can be read in with an additional redirected input file, where the input is grouped into named sets for each heat structure with the film-tracking model switch set to ON. The data for such heat structure may be placed in the additional input file after the character string BLOCK: followed by the user-defined name of this block. To use these data, an INCLUDE directive with the name of

the additional input file and the block name must be input in the original input set after the HS_FT record.

In the following example, an additional input file 'hsft.inp' is used to describe the film tracking of HS40022. The original input file might look like

```
HS_INPUT
!
...
!*** HS data block - 'HS40022' *******************
HS_ID 'HS40022'
! general, source, node, boundaries data records
...
!
! film-tracking model data
HS_FT ON   ! film-tracking model on/off switch
INCLUDE 'hsft.inp' HS40022
!
...
!** END HS INPUT ******************************
```

The file 'hsft.inp' includes the corresponding data block for this heat structure. The following is an example of the input set for two heat structures combined to form a network:

```
! hsft.inp
...
!************************************************
! network xxx
!
BLOCK: HS40022 **********
!
! list of HSs this structure drains to
HS_FTDRN 1 ! NUMDRN IDSDRN   FDRNL   FDRNR
             1      HS40019  1.0     1.0
! data for left  surface
HS_FTLBF 0.0 ! frac of tot drainage from surface
HS_FTLBM NO  ! external mass source rate
HS_FTLBE NO  ! external specific enthalpy source
! data for right surface
HS_FTRBF 0.0 ! frac of tot drainage from surface
HS_FTRBM NO  ! external mass source rate
HS_FTRBE NO  ! external specific enthalpy source
...
!*** END BLOCK: HS40022 ***
!
BLOCK: HS40019 ***********
!
! list of HSs this structure drains to – none
! bottom-most heat structure
HS_FTDRN 0 ! NUMDRN
! data for left  surface
HS_FTLBF 0.0 ! frac of tot drainage from surface
HS_FTLBM NO  ! external mass source rate
HS_FTLBE NO  ! external specific enthalpy source
! data for right surface
HS_FTRBF 0.0
HS_FTRBM NO
HS_FTRBE NO
!** END BLOCK: HS40019 ***
```

```
!
! end network xxx data
!**********************************************
!
```

The user may use comments and sort the blocks in the order of drainage to make a network representation more obvious.

2.1.4 MELGEN Input for Structure-to-Structure Radiation

Input data for the structure-to-structure radiation model are input on the HS_RD record in the form of a table. Each row gives information for a given pair of surfaces. Each surface pair must be unique in that the surfaces associated with one pair must not be associated with another. The number of allowable surface pairs is not restricted; it is limited only by the amount of computer storage available to the database.

HS_RD – Structure-to-Structure Radiation

Optional

(1) NUMPAIR

Number of allowable surface pairs.

(type = integer, default = none, units = none)

Next data are input as a table with number or rows = NUMPAIR:

(2) N

Table entry index.

(type = integer, default = none, units = none)

(3) IHSRD1

Heat structure name for the first surface of the pair.

(type = character*16,default = none)

(4) LRBND1

Option to identify the side of surface IHSRD1.

(a) -1 or LEFT

Left side surface of the given heat structure.

(b) 1 or RIGHT

Right side surface of the given heat structure.

(type = integer / character*5, default = none, units = none)

(5) IHSRD2

Heat structure name for the second surface of the pair.

(type = character*16,default = none)

(6) LRBND2

Option to identify the side of surface IHSRD2.

(a) -1 or LEFT
Left side surface of the given heat structure.

(b) 1 or RIGHT
Right side surface of the given heat structure.

(type = integer / character*5, default = none, units = none)

(7) VIEW

View factor between surface 1 and surface 2.

It must lie in the range of 0.0 to 1.0.

(type = real, default = none, units = none)

(8) ICFRD1

Optional real-valued control function name whose value is the emissivity of surface 1.

(type = character*16, default = NO [see below])

(9) ICFRD2

Optional real-valued control function name whose value is the emissivity of surface 2.

(type = character*16, default = NO [see below])

The input of NO or blank for the control function indices specifies the default use of the COR Package Reference Manual relation (Equation. 2.10) for determining the emissivity as a function of temperature.

The user should note that use of this model is restricted to the surfaces of structures for which the boundary condition flags (IBCL on record HS_LB and IBCR on record HS_RB) do not specify either temperature or heat flux. Thus, the options 0, 20, 30, 80, and 90 are not permitted for IBCL or IBCR. All other options are permitted.

If control functions are used, the emissivity is bounded between 0.0 and 1.0, inclusive, before its use in determining the radiative exchange. The radiative exchange is zeroed if either of the surface emissivities is zero, if the input view factor is zero, or if either surface of the pair is covered by a water pool in its adjacent CVH volume.

Examples

```
HS_RD 4!N IHSRD1        LRBND1 IHSRD2       LRBND1 VIEW ICFRD1   ICFRD2
 1 HeatSlab_01    1     HeatSlab_02   -1    1       CF1     TF22
 2 HeatSlab51501 -1     HeatSlab51502  1    0       NO        NO
 3 HS324         -1     Hs_001         1    1 CLEANING_ON/OFF NO
 4 HS333          1     HS4444        -1    2       NO       CF2
HS_RD 0
```

2.2 MELCOR User Input

This section discusses MELCOR user input for the HS package. All HS package input records are optional for MELCOR execution. Input records are supplied for MELCOR execution only to make permitted changes to the existing database for the HS package. MELCOR input, and these changes are discussed as follows:

Section 2.2.1 Heat structure input records.
Section 2.2.2 Gas source input records.
Section 2.2.3 Film-tracking input records.
Section 2.2.4 MELCOR input for structure-to-structure radiation.

2.2.1 MELCOR Input for Heat Structures

All HS package input records (except for structure-to-structure radiation) are optional for MELCOR execution. Input records for heat structures are supplied for MELCOR execution. Any change to the data for a given heat structure is permitted except one which changes the geometry (HS_GD), or one that involves changes in the number of temperature nodes for the given heat structure. At present a heat structure may not be added or deleted on a restart. All changes will be propagated through subsequent restarts unless rescinded by their input records.

The following items pertaining to input processing by the HS package during a restart are important, and all MELCOR input for heat structures must conform to them:

(1) Although a complete set of records does not need to be supplied for each heat structure with data changes during MELCOR execution, each included record must be complete as described in Section 2.1.1. Thus, all fields on a record preceding and following the field that contains the datum to be changed must be present.

One exception to this is modifying the boundary conditions for a heat structure on a restart. If it is desired to change any of the fields on either the HS_LB (LBP, LBT, LBR) or the HS_RB (RBP, RBT, RBR) boundary surface data records on a restart, then <u>both</u> the HS_LB (LBP, LBT, LBR) and the HS_RB (RBP, RBT, RBR) records for that heat structure must be included in the MELCOR input.

(2) All fields on each record must conform to the criteria or restrictions described in Section 2.1.1.

2.2.2 MELCOR Input for Gas Sources

All HS package gas source input records are optional for MELCOR execution. Input records for gas sources are supplied for MELCOR execution only to make changes to time-independent data on a restart. Any change to the data for a given gas source is permitted. At present, a gas source may not be added or deleted on a restart. Gas source NAMESRC may be made effectively inactive on a restart by setting its source density (RHOSRC on record HS_DGSRC) to zero. All changes will be propagated through subsequent restarts unless rescinded by their input records.

The following items pertaining to input processing by the HS package during a restart are important and must be conformed to by all MELCOR input for gas sources:

(1) Although a complete set of records does not need to be supplied for each gas source with data changes during MELCOR execution, each included record must be complete (see Section 2.1.2 for details). Thus, all fields on a record preceding and following the field that contains the data to be changed must be present.

(2) All fields on each record must conform to the criteria or restrictions as described in Section 2.1.2.

(3) All fields identified by input record HS_DG are not permitted to change for SS degassing sources.

2.2.3 MELCOR Film-Tracking Model Input

All HS package film-tracking model input records are optional for MELCOR execution. Input records for the film-tracking model are supplied for MELCOR execution only to make changes to time-independent data on a restart. The only data that may be changed are values of FRAINL, NTPL, FRAINR, NTPR, IMSRCL, IESRCL, IMSRCR, IESRCR, FDRNL and FDRNR. Networks and the heat structures included in them may not be added or deleted. Restart changes are primarily provided to alter drainage patterns.

Input record HS_FTDRN defined for MELGEN, may also be included in MELCOR input but is not required. Note that the actual number of heat structures to have their values of FDRNL and/or FDRNR modified must be input as the dimension of the table instead of NUMDRN (this value remains unchangeable).

2.2.4 MELCOR Input Controlling Structure-to-Structure Radiation

Although no MELCOR input is allowed for the structure-to-structure radiation model, the model may be effectively turned on and off for individual radiation pairs by using the control function method of specifying surface emissivities (ICFRD1 and ICFRD2) on MELGEN input record HS_RD. For periods during a MELCOR calculation when the calculation of radiation is not desired (between given surfaces previously active in the radiation model), the user can structure the control function such that a zero emissivity—and, thus, zero radiation heat transfer—are produced. Because control function input is allowed upon MELCOR restarts, this method allows user control of the radiation model during all parts of the MELCOR calculation.

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. See Section 7 of the MELCOR EXEC Users' Guide for a description of their use.

3.1 HS Sensitivity Coefficients

The sensitivity coefficients for the HS package have identifier numbers between 4000 and 4299. These are reserved as follows:

4000 – 4049:	Analytic Functions
4050 – 4059:	Iteration Parameters
4060 – 4079	Parameter Ranges for Atmosphere Heat Transfer
4080 – 4099:	Parameter Ranges for Pool Heat Transfer
4100 – 4149:	Atmosphere Heat Transfer Correlations
4150 – 4199:	Pool Heat Transfer Correlations
4200 – 4249:	Mass Transfer Parameters and Correlations
4250 – 4299:	Liquid Film Parameters and Correlations

This section describes the sensitivity coefficients presently used in the HS package and gives their default values, units, and equivalence names. See the MELCOR/MELGEN EXEC Package Users' Guide for details on the use of sensitivity coefficient records.

Warning: In the present version of the MELCOR Code, no error testing of the new value of a sensitivity coefficient is performed in the HS package. Therefore, the user is warned that the indiscriminate change of a coefficient may result in nonphysical results or an aborted calculation.

4000 – Surface Tension

The surface tension of water, σ, is given as a function of temperature by

$$\sigma(T) = C4000(1)x[C4000(2) + C4000(3)xT_R]xT_R^{C4000(4)} + C4000(5)$$

where

$\sigma(T)$	=	surface tension, N/m
T	=	temperature, K
T_R	=	C4000(6) – T/C4000(7)

C4000(1)	Constant coefficient. (default = 0.2358, units = N/m, equiv = none)
C4000(2)	Additive constant. (default = 1.0, units = none, equiv = none)
C4000(3)	Constant coefficient. (default = -0.625, units = none, equiv = none)
C4000(4)	Constant coefficient. (default = 1.256, units = none, equiv = none)

C4000(5)	Additive constant. (default = 0.0, units = N/m, equiv = none)
C4000(6)	Additive constant. (default = 1.0, units = none, equiv = none)
C4000(7)	Critical temperature for water. (default = 647.3, units = K, equiv = none)

4051 – Steady-State Iteration Parameters

These coefficients are iteration parameters that are used for steady-state heat conduction calculations.

C4051(1)	Maximum number of permitted steady-state iterations. (default = 400.0, units = none, equiv = XITMXS)
C4051(2)	Desired relative error tolerance for steady-state calculations; iterations will be performed until the relative error in the temperature profile is less than this value or until XITMXS iterations have been performed [see C4051(5) blow]. (default = 1.0×10^{-5}, units = none, equiv = ERRSS)
C4051(3)	Initial steady-state timestep. (default = 1.0×10^{5}, units = s, equiv = DTSS)
C4051(4)	Maximum relative error tolerance for film mass. (default = 1.0×10^{-2}, units = none, equiv = ERFSS)
C4051(5)	Maximum relative error tolerance for steady-state calculations; failure will be declared if the relative error in the temperature profile exceeds this value after XITMXS iterations. (default = 1.0×10^{-2}, units = none, equiv = ERDIES)

4052 – Steady-State Iteration Weight Parameters

These coefficients are iteration weight parameters that are used for steady-state heat conduction calculations to mitigate nonconvergence by oscillation. See Section 3.1 of the HS Reference Manual for details.

4052(1)	Steady-state boiling heat transfer coefficient weight. (default = 0.0, units = none, equiv = WTBSST)
4052(2)	Error tolerance for steady-state boiling heat transfer coefficient. (default = 0.05, units = none, equiv = ERBSST)

4055 – Transient Iteration Parameters

These coefficients are iteration parameters used for transient conduction calculations.

C4055(1)	Maximum number of permitted transient iterations (must be at least 2.0). (default = 30.0, units = none, equiv = XITMAX)
C4055(2)	Desired relative error tolerance for transient conduction calculations. NOTE: The conduction calculation is normally declared converged when the maximum relative error in the temperature profile within the structure is less than this value. However, if degassing or mass transfer (condensation/ evaporation) is occurring, then the iteration continues until the maximum relative error in the temperature profile (including the film surfaces) is less than the value specified by C4055(6) described below. If the relative error is still larger than C4055(6) but smaller than C4055(2) after XITMAX iterations, then the solution is accepted as converged (see C4055(8) below). (default = 0.5, units = none, equiv = ERRTRN)
C4055(3)	Minimum number of iterations to limit timestep. (default = 31.0, units = none, equiv = XITCUT)
C4055(4)	Minimum relative error tolerance for material property determination. (default = 0.01, units = none, equiv = RETMIN)
C4055(5)	Matrix solver precision limit. (default = 1.0×10^{-10}, units = none, equiv = EPSILL)
C4055(6)	Override value for ERRTRN during degassing/mass transfer—[see the NOTE for C4055(2) above]. (default = 5.0×10^{-6}, units = none, equiv = ERRSML)
C4055(7)	Maximum relative error tolerance for film mass. (default = 1.0×10^{-2} units = none, equiv = ERFTRN)
C4055(8)	Maximum relative error tolerance for transient calculations; failure will be declared and a request made to repeat the cycle with a smaller timestep if the relative error in the temperature profile exceeds this value after XITMAX iterations. (default = 5.0×10^{-3}, units = none, equiv = ERDIET)

4056 – Transient Iteration Weight Parameters

These coefficients are iteration weight parameters that are used for transient heat conduction calculations to mitigate nonconvergence by oscillation. See Section 3.1 of the HS Reference Manual for details.

C4056(1)	Transient boiling heat transfer coefficient weight. (default = 0.9, units = none, equiv = WTBTRN)
C4056(2)	Error tolerance for the transient boiling heat transfer coefficient. (default = 0.05, units = none, equiv = ERBTRN)

4060 – Atmosphere Natural and Forced Convection Ranges

These coefficients define the limits of natural and forced convection heat transfer to the atmosphere. In general, a mixed convection regime is also considered, and the following criteria determine the convection heat transfer to the atmosphere:

Natural convection if $Re^2 <$ C4060(1)xGr
Forced convection if $Re^2 >$ C4060(2)xGr
Mixed convection if C4060(1)x$Gr \leq Re^2 \leq$ C4060(2)xGr

where

Re = Reynolds number for atmosphere
Gr = Grashof number for atmosphere.

However, if C4060(1) < 0 or C4060(2) ≤ C4060(1), convection heat transfer to the atmosphere is assumed to be given by the greater of the values defined by the natural and forced convection correlations.

C4060(1)	Constant coefficient. (default = 1.0, units = none, equiv = none)
C4060(2)	Constant coefficient. (default = 10.0, units = none, equiv = none)

4061 – **4063** – Atmosphere Laminar and Turbulent Natural Convection Ranges

These coefficients define the limits of laminar and turbulent natural convection heat transfer to the atmosphere. In general, a transition regime is also considered, and the following criteria determine the natural convection heat transfer to the atmosphere:

Laminar if $Ra <$ C406m(1)
Turbulent if $Ra >$ C406m(2)
Transition if C406m(1) $\leq Ra \leq$ C406m(2)

where

Ra	=	Rayleigh number for atmosphere
m	=	1 for rectangular geometries
	=	2 for cylindrical geometries
	=	3 for spherical or hemispherical geometries

However, if C406m(1) < 0 or C406m(2) ≤ C406m(1), natural convection heat transfer to the atmosphere is assumed to be given by the greater of the values defined by the laminar and turbulent natural convection correlations.

<u>Rectangular</u>

C4061(1)	Rayleigh number upper limit for atmosphere laminar natural convection. (default = 10^9, units = none, equiv = none)
C4061(2)	Rayleigh number lower limit for atmosphere turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

<u>Cylindrical</u>

C4062(1)	Rayleigh number upper limit for atmosphere laminar natural convection. (default = 10^9, units = none, equiv = none)
C4062(2)	Rayleigh number lower limit for atmosphere turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

<u>(Hemi)spherical</u>

C4063(1)	Rayleigh number upper limit for atmosphere laminar natural convection. (default = 10^9, units = none, equiv = none)
C4063(2)	Rayleigh number lower limit for atmosphere turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

4064 – **4066** – Atmosphere Laminar and Turbulent Forced Convection Ranges

These coefficients define the limits of laminar and turbulent forced convection heat transfer to the atmosphere. In general, a transition regime is also considered, and the following criteria determine the forced convection heat transfer to the atmosphere :

Laminar if *Re* < C406m(1)
Turbulent if *Re* > C406m(2)
Transition if C406m(1) ≤ *Re* ≤ C406m(2)

where

m = 4 for rectangular geometries
= 5 for cylindrical geometries
= 6 for spherical or hemispherical geometries

However, if C406m(1) < 0 or C406m(2) ≤ C406m(1), forced convection heat transfer to the atmosphere is assumed to be given by the greater of the values defined by the laminar and turbulent forced convection correlations.

<u>Rectangular</u>

C4064(1)	Reynolds number upper limit for atmosphere laminar forced convection. (default = $3x10^5$, units = none, equiv = none)
C4064(2)	Reynolds number lower limit for atmosphere turbulent forced convection. (default = 6x105, units = none, equiv = none)

<u>Cylindrical</u>

C4065(1)	Reynolds number upper limit for atmosphere laminar forced convection. (default = 2000, units = none, equiv = none)
C4065(2)	Reynolds number lower limit for atmosphere turbulent forced convection. (default = 10^4, units = none, equiv = none)

<u>(Hemi)spherical</u>

C4066(1)	Reynolds number upper limit for atmosphere laminar forced convection. (default = 2000, units = none, equiv = none)
C4066(2)	Reynolds number lower limit for atmosphere turbulent forced convection. (default = 10^4, units = none, equiv = none)

4071 – Bounds on Critical Pool Fractions

Heat transfer from a heat structure surface to the pool and/or atmosphere of an adjacent control volume is controlled by user input of critical pool fractions on the

HS_LBP and HS_RBP input records. See Section 2.4 of the HS Package Reference Manual. These sensitivity coefficients impose bounds on the user-input values, to prevent transfer of heat to insignificant fluid masses.

C4071(1)	Lower bound imposed on input values of CPFPL and CFFPR. (default = 0.02, units = none, equiv = none)
C4071(2)	Upper bound imposed on input values of CPFAL and CFPAR. (default = 0.98, units = none, equiv = none)

4080 – Pool Natural and Forced Convection Ranges

These coefficients define the limits of natural and forced convection heat transfer to the pool. In general, a mixed convection regime is also considered, and the following criteria determine the convection heat transfer to the pool:

Natural convection if $Re^2 <$ C4080(1)xGr
Forced convection if $Re^2 >$ C4080(2)xGr
Mixed convection if C4080(1)x$Gr \leq Re^2 \leq$ C4080(2)xGr

where

Re =	Reynolds number for pool
Gr =	Grashof number for pool

However, if C4080(1) < 0 or C4080(2) ≤ C4080(1), convection heat transfer to the pool is assumed to be given by the greater of the values defined by the natural and forced convection correlations.

C4080(1)	Constant coefficient. (default = 1.0, units = none, equiv = none)
C4080(2)	Constant coefficient. (default = 10.0, units = none, equiv = none)

4081 – **4083** – Pool Laminar and Turbulent Natural Convection Ranges

These coefficients define the limits of laminar and turbulent natural convection heat transfer to the pool. In general, a transition regime is also considered, and the following criteria determine the natural convection heat transfer to the pool:

Laminar if $Ra <$ C408m(1)
Turbulent if $Ra >$ C408m(2)
Transition if C408m(1) $\leq Ra \leq$ C408m(2)

where

Ra =	Rayleigh number for the pool
m =	1 for rectangular geometries
=	2 for cylindrical geometries
=	3 for spherical or hemispherical geometries

However, if C408m(1) < 0 or C408m(2) ≤ C408m(1), natural convection heat transfer to the pool is assumed to be given by the greater of the values defined by the laminar and turbulent natural convection correlations.

<u>Rectangular</u>

C4081(1)	Rayleigh number upper limit for pool laminar natural convection. (default = 10^9, units = none, equiv = none)
C4081(2)	Rayleigh number lower limit for pool turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

<u>Cylindrical</u>

C4082(1)	Rayleigh number upper limit for pool laminar natural convection. (default = 10^9, units = none, equiv = none)
C4082(2)	Rayleigh number lower limit for pool turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

<u>(Hemi)spherical</u>

C4083(1)	Rayleigh number upper limit for pool laminar natural convection. (default = 10^9, units = none, equiv = none)
C4083(2)	Rayleigh number lower limit for pool turbulent natural convection. (default = 10^{10}, units = none, equiv = none)

4084 – **4086** – Pool Laminar and Turbulent Forced Convection Ranges

These coefficients define the limits of laminar and turbulent forced convection heat transfer to the pool. In general, a transition regime is also considered, and the following criteria determine the forced convection heat transfer to the pool:

Laminar if Re < C408m(1)
Turbulent if Re > C408m(2)
Transition if C408m(1) ≤ Re ≤ C408m(2)

where

m = 4 for rectangular geometries
= 5 for cylindrical geometries
= 6 for spherical or hemispherical geometries

However, if C408m(1) < 0 or C408m(2) ≤ C408m(1), forced convection heat transfer to the pool is assumed to be given by the greater of the values defined by the laminar and turbulent forced convection correlations.

<u>Rectangular</u>

C4084(1) Reynolds number upper limit for pool laminar forced convection.
(default = 3×10^5, units = none, equiv = none)

C4084(2) Reynolds number lower limit for pool turbulent forced convection.
(default = 6×10^5, units = none, equiv = none)

<u>Cylindrical</u>

C4085(1) Reynolds number upper limit for pool laminar forced convection.
(default = 2000, units = none, equiv = none)

C4085(2) Reynolds number lower limit for pool turbulent forced convection.
(default = 10^4, units = none, equiv = none)

<u>(Hemi)spherical</u>

C4086(1) Reynolds number upper limit for pool laminar forced convection.
(default = 2000, units = none, equiv = none)

C4086(2) Reynolds number lower limit for pool turbulent forced convection.
(default = 10^4,units = none, equiv = none)

4101 – **4112** – Atmosphere Natural Convection

The atmosphere natural convection heat transfer correlations have the following form:

$$Nu = C41mm(1)\ x\ Ra^{C41mm(2)} + C41mm(3)$$

where

Nu	=	Nusselt number
Ra	=	Rayleigh number
mm	=	01 (04) for laminar (turbulent) correlations for rectangular geometries in internal flow
	=	02 (05) for laminar (turbulent) correlations for cylindrical geometries in internal flow
	=	03 (06) for laminar (turbulent) correlations for spherical or hemispherical geometries in internal flow
	=	07 (10) for laminar (turbulent) correlations for rectangular geometries in external flow
	=	08 (11) for laminar (turbulent) correlations for cylindrical geometries in external flow
	=	09 (12) for laminar (turbulent) correlations for spherical or hemispherical geometries in external flow

The default values of the coefficients for the atmosphere natural convection heat transfer correlations are summarized in Table 3.1. All are dimensionless and have no assigned equivalence names.

Table 3.1 Default Values for Sensitivity Coefficients 4101 – 4124

Atmosphere Related					
Array	**Type of Flow**	**1**	**2**	**3**	**4**
C4101	L,I,R,NC	0.046	1/3	0	-
C4102	L,I,C,NC	0.046	1/3	0	-
C4103	L,I,S,NC	0.228	0.226	0	-
C4104	T,I,R,NC	0.046	1/3	0	-
C4105	T,I,C,NC	0.046	1/3	0	-
C4106	T,I,S,NC	0.228	0.226	0	-
C4107	L,E,R,NC	0.59	0.25	0	-
C4108	L,E,C,NC	0.59	0.25	0	-
C4109	L,E,S,NC	0.43	0.25	2.0	-
C4110	T,E,R,NC	0.10	1/3	0	-
C4111	T,E,C,NC	0.10	1/3	0	-
C4112	T,E,S,NC	0.43	0.25	2.0	-
C4113	L,I,R,FC	8.235	0	0	0
C4114	L,I,C,FC	48/11	0	0	0

Atmosphere Related					
Array	**Type of Flow**	**1**	**2**	**3**	**4**
C4115	L,I,S,FC	48/11	0	0	0
C4116	T,I,R,FC	0.023	0.8	1/3	0
C4117	T,I,C,FC	0.023	0.8	1/3	0
C4118	T,I,S,FC	0.023	0.8	1/3	0
C4119	L,E,R,FC	0.664	0.5	1/3	0
C4120	L,E,C,FC	0.664	0.5	1/3	0
C4121	L,E,S,FC	0.60	0.5	1/3	2.0
C4122	T,E,R,FC	0.037	0.8	1/3	0
C4123	T,E,C,FC	0.037	0.8	1/3	0
C4124	T,E,S,FC	0.60	0.5	1/3	2.0

L=Laminar, T=Turbulent, I=Internal, E=External, R=Rectangular, C=Cylindrical, S=(Hemi)spherical, NC=Natural Convection, FC=Forced Convection

4113 – **4124** – Atmosphere Forced Convection

The atmosphere forced convection heat transfer correlations have the following form:

$$Nu = C41mm(1) x \, \mathrm{Re}^{C41mm(2)} \, x \, \mathrm{Pr}^{C41mm(3)} + C41mm(4)$$

where

Nu	=	Nusselt number
Re	=	Reynolds number
Pr	=	Prandtl number
mm	=	13 (16) for laminar (turbulent) correlations for rectangular geometries in internal flow
	=	14 (17) for laminar (turbulent) correlations for cylindrical geometries in internal flow
	=	15 (18) for laminar (turbulent) correlations for spherical or hemispherical geometries in internal flow
	=	19 (22) for laminar (turbulent) correlations for rectangular geometries in external flow
	=	20 (23) for laminar (turbulent) correlations for cylindrical geometries in external flow
	=	21 (24) for laminar (turbulent) correlations for spherical or hemispherical geometries in external flow

The default values of the coefficients for the atmosphere forced convection heat transfer correlations are summarized in Table 3.1. All are dimensionless and have no assigned equivalence names.

4151 – **4162** – Pool Natural Convection

The pool natural convection heat transfer correlations have the following form:

$$Nu = C41mm(1) x Ra^{C41mm(2)} + C41mm(3)$$

where

Nu	=	Nusselt number
Ra	=	Rayleigh number
mm	=	51 (54) for laminar (turbulent) correlations for rectangular geometries in internal flow
	=	52 (55) for laminar (turbulent) correlations for cylindrical geometries in internal flow
	=	53 (56) for laminar (turbulent) correlations for spherical or hemispherical geometries in internal flow
	=	57 (60) for laminar (turbulent) correlations for rectangular geometries in external flow
	=	58 (61) for laminar (turbulent) correlations for cylindrical geometries in external flow
	=	59 (62) for laminar (turbulent) correlations for spherical or hemispherical geometries in external flow.

The default values of the coefficients for the pool natural convection heat transfer correlations are summarized in Table 3.2. All are dimensionless and have no assigned equivalence names.

Table 3.2 Default Values for Sensitivity Coefficients 4151 – 4174

Pool Related					
Array	**Type of Flow**	**1**	**2**	**3**	**4**
C4151	L,I,R,NC	0.046	1/3	0	-
C4152	L,I,C,NC	0.046	1/3	0	-
C4153	L,I,S,NC	0.228	0.226	0	-
C4154	T,I,R,NC	0.046	1/3	0	-
C4155	T,I,C,NC	0.046	1/3	0	-
C4156	T,I,S,NC	0.228	0.226	0	-
C4157	L,E,R,NC	0.59	0.25	0	-

Pool Related					
Array	**Type of Flow**	**1**	**2**	**3**	**4**
C4158	L,E,C,NC	0.59	0.25	0	-
C4159	L,E,S,NC	0.43	0.25	2.0	-
C4160	T,E,R,NC	0.10	1/3	0	-
C4161	T,E,C,NC	0.10	1/3	0	-
C4162	T,E,S,NC	0.43	0.25	2.0	-
C4163	L,I,R,FC	8.235	0	0	0
C4164	L,I,C,FC	48/11	0	0	0
C4165	L,I,S,FC	48/11	0	0	0
C4166	T,I,R,FC	0.023	0.8	1/3	0
C4167	T,I,C,FC	0.023	0.8	1/3	0
C4168	T,I,S,FC	0.023	0.8	1/3	0
C4169	L,E,R,FC	0.664	0.5	1/3	0
C4170	L,E,C,FC	0.664	0.5	1/3	0
C4171	L,E,S,FC	0.60	0.5	1/3	2.0
C4172	T,E,R,FC	0.037	0.8	1/3	0
C4173	T,E,C,FC	0.037	0.8	1/3	0
C4174	T,E,S,FC	0.60	0.5	1/3	2.0

L=Laminar, T=Turbulent, I=Internal, E=External, R=Rectangular, C=Cylindrical, S=(Hemi)spherical, NC=Natural Convection, FC=Forced Convection

4163 – **4174** – Pool Forced Convection

The pool forced convection heat transfer correlations have the following form:

$$Nu = C41mm(1)x\,\mathrm{Re}^{C41mm(2)}\,x\,\mathrm{Pr}^{C41mm(3)} + C41mm(4)$$

where

Nu	=	Nusselt number
Re	=	Reynolds number
Pr	=	Prandtl number
mm	=	63 (66) for laminar (turbulent) correlations for rectangular geometries in internal flow
	=	64 (67) for laminar (turbulent) correlations for cylindrical geometries in internal flow
	=	65 (68) for laminar (turbulent) correlations for spherical or

hemispherical geometries in internal flow
= 69 (72) for laminar (turbulent) correlations for rectangular geometries in external flow
= 70 (73) for laminar (turbulent) correlations for cylindrical geometries in external flow
= 71 (74) for laminar (turbulent) correlations for spherical or hemispherical geometries in external flow.

The default values of the coefficients for the pool forced convection heat transfer correlations are summarized in Table 3.2. All are dimensionless and have no assigned equivalence names.

4180 – Nucleate Boiling Heat Flux

The nucleate boiling heat flux is obtained through the Rohsenow relation

$$\left[\frac{C_{pl}\left(T_{surf}-T_{sat}\right)}{h_{fg}}\right]=C4180(1)x\,\mathrm{Pr}^{C4180(3)}\,x\left[\frac{q''_{NB}}{\mu h_{fg}}\left\{\frac{\sigma}{g\left(\rho_l-\rho_v\right)}\right\}^{1/2}\right]^{C4180(4)}$$

where

q''_{NB}	=	Nucleate boiling heat flux, W/m^2
c_{pl}	=	Heat capacity of liquid at T_{sat}, J/kg-K
T_{surf}	=	Temperature of the surface, K
T_{sat}	=	Saturation temperature in the boundary volume, K
h_{fg}	=	Latent heat in the boundary volume of this surface, J/kg
Pr	=	Prandtl number of liquid in the boundary volume
μ	=	Dynamic viscosity of liquid at T_{avg}, kg/m-s
T_{avg}	=	(T_{surf} + T_{sat})/2, K
σ	=	Surface tension at T_{avg}, N/m
g	=	Acceleration of gravity, m/s^2
ρ_l	=	Density of liquid at T_{sat}, kg/m^3
ρ_v	=	Density of vapor at T_{sat}, kg/m^3

C4180(1)	Constant determined empirically for different surfaces and fluids. (default = 0.013, units = none, equiv = none)
C4180(2)	Unused.
C4180(3)	Prandtl number exponent. (default = 1.0, units = none, equiv = none)
C4180(4)	Exponent. (default = 0.33, units = none, equiv = none)

4181 – Critical Heat Flux

The critical heat flux is given by Zuber as

$$q''_c = C4181(1) x \rho_v h_{fg} \left[\sigma(\rho_l - \rho_v) g / \rho_v^2\right]^{C4181(2)} x \left[\rho_l / (\rho_l + \rho_v)\right]^{C4181(3)}$$

where

q''_c	=	Critical heat flux, W/m^2
ρ_v	=	Density of vapor at T_{sat}, kg/m^3
ρ_l	=	Density of liquid at T_{sat}, kg/m^3
h_{fg}	=	Latent heat in the boundary volume, J/kg
g	=	Acceleration of gravity, m/s^2
σ	=	Surface tension at T_{avg}, N/m
T_{avg}	=	(T_{surf} + T_{sat}) / 2, K
T_{sat}	=	Saturation temperature in the boundary volume, K
T_{surf}	=	Temperature of this surface, K

C4181(1) Constant coefficient.
(default = 0.18, units = none, equiv = none)

C4181(2) Exponent.
(default = 0.25, units = none, equiv = none)

C4181(3) Exponent.
(default = 0.5, units = none, equiv = none)

4182 – Minimum Film Boiling Heat Flux

The minimum film boiling heat flux is given by Zuber as

$$q''_{mfilm} = C4182(1) \rho_v h_{fg} \left[\sigma(\rho_l - \rho_v) g / \rho_l^2\right]^{C4182(2)} x \left[\rho_l (\rho_l + \rho_v)\right]^{C4182(3)}$$

where

q''_{mfilm}	=	minimum film boiling heat flux, W/m^2

C4182(1) Constant coefficient.
(default = 0.09, units = none, equiv = none)

C4182(2) Exponent.
(default = 0.25, units = none, equiv = none)

C4182(3) Exponent.
(default = 0.5, units = none, equiv = none)

4183 – Film Boiling Heat Flux

The film boiling heat flux is given by Bromley as

$$q''_{film} = C4183(1)x\left[\rho_v(\rho_l - \rho_v)g\ k_v^3\left(h_{fg} + \frac{1}{2}c_{pv}\Delta T\right) / \mu_v L_c\right]^{C4183(2)} x\Delta T^{C4183(3)}$$

where

q''_{film}	=	Film boiling heat flux, W/m^2
L_c	=	Characteristic length of this surface, m
ΔT	=	$T_{surf} - T_{sat}$, K
T_{surf}	=	Temperature of this surface, K
T_{sat}	=	Saturation temperature in the boundary volume, K
g	=	Acceleration due to gravity, m/s^2
h_{fg}	=	Latent heat in the boundary volume, J/kg
ρ_l	=	Density of liquid at T_{sat}, kg/m^3
ρ_v	=	Density of vapor at T_{sat}, kg/m^3
c_{pv}	=	Heat capacity of vapor at T_{sat}, J/kg-K
T_{avg}	=	(T_{surf} + T_{sat}) / 2, K
μ_v	=	Dynamic viscosity of vapor at T_{avg}, kg/m-s
k_v	=	Thermal conductivity of vapor at T_{avg}, W/m-K

C4183(1)	constant coefficient (default = 0.943, units = none, equiv = none)
C4183(2)	exponent (default = 0.25, units = none, equiv = none)
C4183(3)	exponent (default = 0.75, units = none, equiv = none)

4184 – Radiation to Pool Heat Flux (Plane Model)

The radiation to pool heat flux is given by

$$q''_{rad} = C4184(1)x\sigma\left(T_{surf}^4 - T_{pool}^4\right)$$

where

q''_{rad}	=	Radiation to pool heat flux, W/m^2
T_{surf}	=	Temperature of the surface, K
T_{pool}	=	Temperature of pool in the boundary volume, K
σ	=	Stefan-Boltzmann constant, 5.669x10^{-8} W/m^2-K^4

C4184(1) Constant coefficient.
(default = 1.0, units = none, equiv = none)

4200 – Mass Transfer Flux Model Transition Parameter

The condensation/evaporation mass flux is not limited by the presence of noncondensible gases if the ratio of the steam partial pressure to the total pressure in the bulk atmosphere exceeds this parameter.

C4200(1) Ratio of steam partial pressure to total pressure in bulk atmosphere.
(default = 0.9995, units = none, equiv = VPFRAC)

4201 – Sherwood Number for Diffusion Mass Transfer

A Sherwood Number Correlation calculates a diffusion mass transfer coefficient. The correlation has the following form:

$$Sh = C4201(1) x Nu^{C4201(2)} x Sc^{C4201(3)} x \Pr^{C4201(4)}$$

where

Nu	=	Nusselt number
Sc	=	Schmidt number
Pr	=	Prandtl number.

C4201(1) Constant coefficient.
(default = 1.0, units = none, equiv = none)

C4201(2) Nusselt number exponent.
(default = 1.0, units = none, equiv = none)

C4201(3) Schmidt number exponent.
(default = 1/3, unit = none, equiv = none)

C4201(4) Prandtl number exponent.
(default = -1/3, units = none, equiv = none)

4202 – Film Flashing Heat Transfer Coefficient

This parameter limits the rate of film evaporation calculated by the mechanistic formula used when noncondensibles are present. The mechanistic formula is

$$\dot{m}_c = h_D \rho_v \ln\left(\Delta P_{srf} / \Delta P_{atm}\right)$$

where

$\dot{m}_c$	=	Mass flux at this surface, kg/m^2-s
h_D	=	Mass transfer coefficient, m/s
ρ_v	=	Density of vapor at $T_{sat}(P_{tot})$, kg/m^3
ΔP_{srf}	=	P_{tot} - P_{srf} = noncondensible partial pressure at the surface temperature, Pa
ΔP_{atm}	=	P_{tot} - P_{stm} = noncondensible partial pressure in the bulk atmosphere, Pa
P_{tot}	=	Total control volume pressure, Pa
P_{srf}	=	Saturation pressure of steam at the surface temperature, Pa
P_{stm}	=	Steam partial pressure in the control volume, Pa.

Because this equation is singular when the P_{srf} reaches P_{tot} it is necessary to bound the rate of evaporation as the surface temperature reaches T_{sat} (P_{tot}). This is done using a flashing heat transfer coefficient to limit the rate of evaporation as follows:

$$\dot{m}_e = h_e \min\left(0, T_{dew} - T_{srf}\right) / h_{fg}$$

where

$$\dot{m} = \max\left(\dot{m}_c, \dot{m}_e\right)$$

h_e	=	Maximum flashing heat transfer coefficient, W/m^2-K
h_{fg}	=	Latent heat of vaporization for steam, J/kg
T_{dew}	=	Control volume dew point temperature, K
T_{srf}	=	Surface temperature, K

and h_e has been implemented as sensitivity coefficient 4202, with a default value of 5.x10^5 W/m^2-K as follows:

C4202(1) Maximum evaporative heat transfer coefficient.
(default = 5.x10^5, units = W/m^2-K, equiv = HTFLSH)

4203 – CVH Steam Depletion Control Parameters

These parameters determine what corrective measure is applied if the calculated condensation rates cause excessive steam depletion in a CVH control volume. Steam depletion is considered excessive if the mass of steam condensed during any timestep exceeds C4203(2) x $M_{stm,0}$, where $M_{stm,0}$ is the initial mass of steam in the control volume at the beginning of the timestep. When excessive condensation occurs, one of two corrective actions is taken. By default, the recommended procedure is to repeat the cycle with a smaller timestep to avoid the problem. The alternative, which is invoked any time the current timestep is less than the value of C4203(1), is to scale the calculated condensation fluxes in the offending control

volume by a calculated factor less than one, which will eliminate the excessive condensation. The scaling method is not recommended, except as a last resort, because it can artificially limit the true rate of condensation. As an alternative to scaling the condensation rate or limiting the timestep, it may be preferable to renodalize the problem to reduce the ratio of the HS structure surface area to the CVH volume in the region of the excessive condensation.

C4203(1)	Fallback/scaling timestep size transition parameter. (default = -1.0, units = s, equiv = DTCOND)
C4203(2)	Maximum steam mass fraction that may condense per timestep. (default = 0.9, units = none, equiv = STMFAC)

4205 – Stainless Steel Melting (Degassing) Parameters

These parameters control the melting of stainless steel heat structures via the degassing model.

C4205(1)	Mass of unmelted steel below which the heat structure is deactivated and the remaining mass is relocated to the COR package. (default = 10., units = kg, equiv = none)

4210 – **4215**, **4220** – **4222**, **4230** – **4232** – Film Heat Transfer Coefficient Correlations

The following criteria determine the laminar or turbulent heat transfer through the condensate film:

Laminar if Re_f < C42m0(1)
Turbulent if Re_f > C42m0(2)
Transition if C42m0(1) $\leq$ Re_f $\leq$ C42m0(2)

where

Re_f = Reynolds number for the film flow.

The laminar heat transfer coefficient through the film, $h_{f,l}$, is given by

$$h_{f,l} = (k_f / L) \ x \ Nu_{f,l}$$

where the laminar film Nusselt number, $Nu_{f,l}$, is given by

$$Nu_{f,l} = C42m1(1) x \left\{ g \ \rho_f (\rho_f - \rho_v) h_{fg} L^3 \sin\theta / \left[\mu_f k_f (T_{sat} - T_{srf}) \right] \right\}^{C42m1(2)} .$$

The turbulent heat transfer coefficient through the film, $h_{f,t}$, is given by

$$h_{f,t} = \left\{ k_f / \left[(\mu_f / \rho_f)^2 / g \right]^{C42m2(1)} \right\} x \;\; Nu_{f,t}$$

where the turbulent film Nusselt number, $Nu_{f,t}$ is given by

$$Nu_{f,t} = \left(\mathrm{Re}_f^{C42m2(2)} + C42m2(3)\mathrm{Re}_f^{C42m2(4)} \mathrm{Pr}_f^{C42m2(5)} \right)^{C42m2(6)}.$$

The transition heat transfer coefficient through the film, $h_{f,tr}$, is given by linear interpolation of Re_f as

$$h_{f,tr} = h'_{f,l} + \left[h'_{f,t} - h'_{f,l} \right] x \left[\mathrm{Re}_f - C42m0(1) \right] / \left[C42m0(2) - C42m0(1) \right].$$

In each of these equations

k_f	=	Thermal conductivity of film
L	=	Characteristic length of surface
ρ_f	=	Density of film
ρ_v	=	Density of vapor
g	=	Acceleration of gravity
h_{fg}	=	Latent heat of vaporization corrected for sensible heat $\left(h_{fg} + 0.68\, c_{p,f} \left(T_f - T_{srf} \right) \right)$
$c_{p,f}$	=	Specific heat capacity of film
T_f	=	Temperature of film/atmosphere interface
T_{srf}	=	Temperature of film/structure interface
μ_f	=	Viscosity of film
θ	=	Angle between the horizontal and the structure surface or axis (for cylindrical geometries)
$h'_{f,l}$	=	$h_{f,l}$ evaluated with Re_f = C42m0(1)
$h'_{f,t}$	=	$h_{f,t}$ evaluated with Re_f = C42m0(2)

and

m	=	1 for upward-facing rectangular geometries
	=	2 for horizontal cylindrical geometries
	=	3 for spherical or hemispherical geometries

For downward-facing rectangular geometries, the laminar/turbulent transition criteria are given by

Laminar if Ra_f < C4213(2)
Turbulent, otherwise

where

Ra_f = Rayleigh number for the film flow

The heat transfer coefficient through the film is given by

$$h_f = \{k_f / [\sigma_f / g(\rho_f - \rho_v)\cos\theta]\}^{1/2} x Nu_f$$

where the film Nusselt number is given by

$$Nu_f = C4214(1) x \{\max[C4213(1), Ra_f]\}^{C4214(2)}$$

for laminar film flow, and

$$Nu_f = C4215(1) x \{\min[C4213(3), Ra_f]\}^{C4215(2)}$$

for turbulent film flow.

<u>Rectangular (upward facing)</u>

C4210(1)	Reynolds number upper limit for laminar film flow (default = 30.0, units = none, equiv = none)
C4210(2)	Reynolds number lower limit for turbulent film flow (default = 100.0, units = none, equiv = none)
C4210(3)	(Do not use.)
C4210(4)	Sine of the minimum angle from the horizontal. (default = 0.1686289, units = none, equiv = none)
C4211(1)	Laminar correlation leading coefficient. (default = 0.943, units = none, equiv = none)
C4211(2)	Laminar correlation exponent. (default = 0.25, units = none, equiv = none)
C4212(1)	Turbulent correlation exponent. (default = 0.3333333, units = none, equiv = none)
C4212(2)	Turbulent correlation exponent. (default = - 0.44, units = none, equiv = none)
C4212(3)	Turbulent correlation coefficient. (default = 5.82×10^{-6}, units = none, equiv = none)
C4212(4)	Turbulent correlation exponent. (default = 0.8, units = none, equiv = none)
C4212(5)	Turbulent correlation exponent. (default = 0.3333333, units = none, equiv = none)
C4212(6)	Turbulent correlation exponent. (default = 0.5, units = none, equiv = none)

Rectangular (downward facing)

C4213(1)	Lower constraint on Ra_f. (default = 10^6, units = none, equiv = none)
C4213(2)	Transition value of Ra_f. (default = 10^8, units = none, equiv = none)
C4213(3)	Upper constraint on Ra_f. (default = 10^{10}, units = none, equiv = none)
C4213(4)	Minimum value of cosine (angle) for use of the downward-facing model. (default = 0.9715642, units = none, equiv = none)
C4214(1)	Laminar correlation leading coefficient. (default = 0.6, units = none, equiv = none)
C4214(2)	Laminar correlation exponent. (default = 0.2, units = none, equiv =none)
C4215(1)	Turbulent correlation leading coefficient. (default = 0.72, units = none, equiv = none)
C4215(2)	Turbulent correlation exponent. (default = 0.19, units = none, equiv = none)

NOTE: To avoid potential numerical difficulties, the laminar and turbulent correlations must be continuous at the transition Ra_f number given by C4213(2).

Horizontal Cylindrical

C4220(1)	Reynolds number upper limit for laminar film flow. (default = 30.0,units = none, equiv = none)
C4220(2)	Reynolds number lower limit for turbulent film flow. (default = 100.0, units = none, equiv = none)
C4220(3)	(Do not use.)
C4220(4)	Cosine of maximum angle from the horizontal. (default = 0.9715642, units = none, equiv = none)

NOTE: If the value of C4220(4) is set equal to 0., then the cylindrical correlations will be applied to all cylinders. As long as the radius of the cylinder is very large compared to the film thickness, use of the correlations, regardless of the angle of inclination, is probably justified. Since many important structures are vertical cylinders (e.g., containment vessels, reactor vessels, core shrouds, and baffles), the matter should be given careful consideration.

C4221(1)	Laminar correlation leading coefficient.

(default = 0.729, units = none, equiv = none)

C4221(2) Laminar correlation exponent.
(default = 0.25, units = none, equiv = none)

C4222(1) Turbulent correlation exponent.
(default = 0.3333333, units = none, equiv = none)

C4222(2) Turbulent correlation exponent.
(default = - 0.44, units = none, equiv = none)

C4222(3) Turbulent correlation coefficient.
(default = 5.82×10^{-6}, units = none, equiv = none)

C4222(4) Turbulent correlation exponent.
(default = 0.8, units = none, equiv = none)

C4222(5) Turbulent correlation exponent.
(default = 0.3333333, units = none, equiv = none)

C4222(6) Turbulent correlation exponent.
(default = 0.5, units = none, equiv = none)

<u>(Hemi)spherical</u>

C4230(1) Reynolds number upper limit for laminar film flow.
(default = 30.0, units = none, equiv = none)

C4230(2) Reynolds number lower limit for turbulent film flow.
(default = 100.0, units = none, equiv = none)

C4231(1) Laminar correlation leading coefficient.
(default = 0.815, units = none, equiv = none)

C4231(2) Laminar correlation exponent.
(default = 0.25, units = none, equiv = none)

C4232(1) Turbulent correlation exponent.
(default = 0.3333333, units = none, equiv = none)

C4232(2) Turbulent correlation exponent.
(default = - 0.44, units = none, equiv = none)

C4232(3) Turbulent correlation coefficient.
(default = 5.82×10^{-6}, units = none, equiv = none)

C4232(4) Turbulent correlation exponent.
(default = 0.8, units = none, equiv = none)

C4232(5) Turbulent correlation exponent.
(default = 0.3333333, units = none, equiv = none)

C4232(6) Turbulent correlation exponent.
(default = 0.5, units = none, equiv = none)

4251 – Minimum and Maximum Liquid Film Thickness

These coefficients represent the minimum and maximum thickness of a liquid film on a surface. The maximum film thickness will be ignored if film tracking is active.

C4251(1)	Constant minimum film thickness. (default = 10^{-9}, units = m, equiv = FILMIN)
C4251(2)	Constant maximum film thickness. (default = 0.0005, units = m, equiv = FILMAX)

4252 – Film Modeling Options

The default treatment of water films on structure surfaces has been revised. These coefficients serve as flags to allow the disabling of the revised models, so that the calculation reverts to that used in MELCOR 1.8.5.

C4252(1)	Rising pool option. If zero, a rising pool surface will subsume the mass of the film that it covers. If non-zero, the film mass will be pushed up as in MELCOR 1.8.5. (default = 0.0, units = none, equiv = none)
C4252(2)	Dynamic draining option. If zero, the dynamic model for film draining used for surfaces in film-tracking networks will be applied to <u>all</u> surfaces and, in effect, each surface not explicitly named on the HS_FTDRN record will be treated as a network of length 1. If non-zero, only the mass that exceeds the quasi-steady thickness derived from the condensation correlations will be drained. (default = 0.0, units = none, equiv = none)

4253 – Film-Tracking Model Correlation Parameters

The film thickness, δ_f, is given as a function of the Reynolds number of the film flow by the following correlation:

$$\delta_t = \begin{cases} C4253(1) \cdot \delta^* \cdot \mathrm{Re}_f^{C4253(2)}, \text{if } \mathrm{Re}_f < C4253(5) \\ C4253(3) \cdot \delta^* \cdot \mathrm{Re}_f^{C4253(4)}, \text{if } \mathrm{Re}_f > C4253(6) \\ \text{Value determined by linear interpolation betwewen boundary limits} \end{cases}$$

and, during film tracking, the heat transfer coefficient through the film is given by

$$h_f = \begin{cases} k_f / \delta_f, \text{if } \mathrm{Re}_f < C4253(5) \\ (k_f / \delta^*)\left[\mathrm{Re}_f^{C4253(7)} + C4253(8)\mathrm{Re}_f^{C4253(9)}\,\mathrm{Pr}_f^{C4253(10)}\right]^{C4253(11)}, \text{if } \mathrm{Re}_f > C4253(6) \\ \text{Value determined by linear interpolation between boundary limits} \end{cases}$$

where

$$\delta^* = \left[(\mu_f \rho_f)^2 / (g \cdot \sin\theta)\right]^{1/3}$$

and k_f, μ_f, ρ_f, and Pr_f are the thermal conductivity, viscosity, density, and Prandtl number of the film, respectively, and θ is the angle of the surface with respect to the horizontal.

C4253(1)	Low Reynolds number film thickness correlation constant. (default = 0.909, units = none, equiv = none)
C4253(2)	Low Reynolds number film thickness correlation exponent. (default = 0.3333333, units = none, equiv = none)
C4253(3)	High Reynolds number film thickness correlation constant. (default = 0.115, units = none, equiv = none)
C4253(4)	High Reynolds number film thickness correlation exponent. (default = 0.6, units = none, equiv = none)
C4253(5)	Low Reynolds number limit. (default = 1000., units = none, equiv = none)
C4253(6)	High Reynolds number limit. (default = 3000., units = none, equiv = none)
C4253(7)	High Reynolds number film heat transfer coefficient exponent. (default = -0.44, units = none, equiv = none)
C4253(8)	High Reynolds number film heat transfer coefficient constant. (default = 5.82×10^{-6}, units = none, equiv = none)
C4253(9)	High Reynolds number film heat transfer coefficient exponent. (default = 0.8, units = none, equiv = none)
C4253(10)	High Reynolds number film heat transfer coefficient exponent. (default = 0.3333333, units = none, equiv = none)

C4253(11) High Reynolds number film heat transfer coefficient exponent.
(default = 0.5, units = none, equiv = none)

1245 – Downward-Facing Lower Head Heat Transfer Correlations

These coefficients calculate downward heat transfer from the lower head to the water in the reactor cavity. The critical heat flux as a function of the inclination of the surface ($\theta = 0°$ for downward-facing surfaces) is given by

$$q_{CHF} = [C1245(1) + C1245(2) \cdot \theta^{C1245(3)}](\rho_v)^{0.5} h_{lv} [g\sigma(\rho_l - \rho_v)]^{0.25}$$

where ρ_l and ρ_v are the densities of water and steam, respectively; σ is the interfacial surface tension between steam and water; g is the acceleration of gravity; and h_{lv} is the latent heat of vaporization of water. The minimum heat flux at the beginning of stable film boiling is given by

$$q_{MIN} = [C1245(4) + C1245(5) \cdot \theta^{C1234(6)}](\rho_v)^{0.5} h_{lv} [g\sigma(\rho_l - \rho_v)]^{0.25}$$

and the film-boiling heat transfer coefficient is given by

$$h_{FLM} = f(\theta) k_v [h_{lv} \rho_v g(\rho_l - \rho_v) / (\mu_v K_v \Delta T)]^{1/3}$$

where k_v is the thermal conductivity of steam, μ_v is the viscosity of steam, and $\Delta T = T_{SURF} - T_{SAT}$ is the difference between the surface temperature and the saturation temperature, and $f(\theta)$ gives the angular dependence. There are two options for calculating $f(\theta)$. If C1245(7)=0., then $f(\theta)$ is given by

$$f(\theta) = C1245(8)(\sin\theta)^{C1245(9)};$$

otherwise, it is given by

$$f(\theta) = C1245(10) + C1245(11) \cdot \theta^{C1245(12)}.$$

See Section 5.1 of the COR Package Reference Manual for details.

(1) Additive constant in the q_{CHF} correlation.
(default = 0.034, units = none, equiv = none)

(2) Multiplicative constant in the q_{CHF} correlation.
(default = 0.0037, units = none, equiv = none)

(3) Exponent of θ in the q_{CHF} correlation.
(default = 0.656, units = none, equiv = none)

(4) Additive constant in the q_{MIN} correlation.
(default = 4.8E-4, units = none, equiv = none)

(5) Multiplicative constant in the q_{MIN} correlation.
(default = 8.2E-4, units = none, equiv = none)

(6) Exponent of θ in the q_{MIN} correlation.
(default = 0.407, units = none, equiv = none)

(7) Film-boiling heat transfer coefficient correlation selection switch.
(default = 0., units = none, equiv = none)

(8) Multiplicative constant in the default film-boiling correlation.
(default = 0.142, units = none, equiv = none)

(9) Exponent of sin θ dependence in the default film-boiling correlation.
(default = 0.3333333, units = none, equiv = none)

(10) Additive constant in the optional film-boiling correlation.
(default = 0.055, units = none, equiv = none)

(11) Multiplicative constant in the optional film-boiling correlation.
(default = 0.016, units = none, equiv = none)

(12) Exponent of θ in the optional film-boiling correlation.
(default = 0.5, units = none, equiv = none)

HS_SC – HS Sensitivity Coefficients Input Record

Optional

This record allows change of the default HS sensitivity coefficient values to user-input ones. This record is not required.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the HS sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only; no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
HS_SC   5   ! N   SC          VALUE     INDEX
              1   4000        0.25          1
              2   4000        1.3           4
              3   4055        33.0          3
              4   4055        1.0E-12       5
              5   4055        1.0E-4        8
```

4 Plot Variables

The plot variables are described in this section. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

HS-CPUC	Total CPU time used for HS package calculations. (s)
HS-CPUE	Total CPU time used for HS package edits. (s)
HS-CPUR	Total CPU time used for HS package restarts. (s)
HS-DEGAS-ENERGY.G	Total energy of released gas from gas source index G. (J)
HS-DEGAS-MASS.G	Total mass of released gas from gas source index G. (kg)
HS-DEGAS-RATE.G	Degassing rate of gas source index G. (kg/s)

HS-DEGAS-STEELM.G	Steel mass melted during the system cycle for steel degassing source index G. (kg)
HS-DEGAS-CSTEELM.G	Cumulative mass of steel melted for steel degassing source associated with index G. (kg)
HS-RAD-FLUX-L.C	Left HS index C to HS radiation heat transfer. (J)
HS-RAD-FLUX-R.C	Right HS index C to HS radiation heat transfer. (J)
HS-DELE-ATMS-L.C	Integrated energy transfer to boundary volume atmosphere of left (inside) boundary surface of heat structure index C due to mass transfer. (J)
HS-DELE-ATMS-R.C	Integrated energy transfer to boundary volume atmosphere of right (outside) boundary surface of heat structure index C due to mass transfer. (J)
HS-DELE-POOL-L.C	Integrated energy transfer to boundary volume pool of left (inside) boundary surface of heat structure index C due to mass transfer. (J)
HS-DELE-POOL-R.C	Integrated energy transfer to boundary volume pool of right (outside) boundary surface of heat structure index C due to mass transfer. (J)
HS-FILM-TEMP-L.C	Temperature of film/atmosphere interface at left boundary surface of heat structure index C. (K)
HS-FILM-TEMP-R.C	Temperature of film/atmosphere interface at right boundary surface of heat structure index C. (K)
HS-DELM-DROP-L.C	Integrated droplet (fog) mass transfer to boundary volume atmosphere of left (inside) boundary surface of heat structure index C.

	(kg)
HS-DELM-DROP-R.C	Integrated droplet (fog) mass transfer to boundary volume atmosphere of right (outside) boundary surface of heat structure index C. (kg)
HS-DELM-POOL-L.C	Integrated water mass transfer to boundary volume pool of left (inside) boundary surface of heat structure index C. (kg)
HS-DELM-POOL-R.C	Integrated water mass transfer to boundary volume pool of right (outside) boundary surface of heat structure index C. (kg)
HS-DELM-STEAM-L.C	Integrated vapor (steam) mass transfer to boundary volume atmosphere of left (inside) boundary surface of heat structure index C. (kg)
HS-DELM-STEAM-R.C	Integrated vapor (steam) mass transfer to boundary volume atmosphere of right (outside) boundary surface of heat structure index C. (kg)
HS-ENERGY-FLUX-L.C	Energy flux associated with HS-MASS-FLUX-L.C. (W/m2)
HS-ENERGY-FLUX-R.C	Energy flux associated with HS-MASS-FLUX-R.C. (W/m2)
HS-ENERGY-INPUT.C	Energy input to heat structure index C by internal and surface power sources and by other packages. (J)
HS-ENERGY-STORED.C	Stored energy of heat structure index C. (J)
HS-FILM-ENTH-L.C	Specific enthalpy of liquid film on left (inside) boundary surface of heat structure index C. (J/kg)

HS-FILM-ENTH-R.C	Specific enthalpy of liquid film on right (outside) boundary surface of heat structure index C. (J/kg)
HS-FILM-MASS-L.C	Mass of liquid film on left (inside) boundary surface of heat structure index C. (kg)
HS-FILM-MASS-R.C	Mass of liquid film on right (outside) boundary surface of heat structure index C. (kg)
HS-FILM-THICK-L.C	Thickness of liquid film on left (inside) boundary surface of heat structure index C. (m)
HS-FILM-THICK-R.C	Thickness of liquid film on right (outside) boundary surface of heat structure index C. (m)
HS-HTC-ATMS-L.C	Atmosphere heat transfer coefficient at left (inside) boundary surface of heat structure index C. (W/m^2-K)
HS-HTC-ATMS-R.C	Atmosphere heat transfer coefficient at right (outside) boundary surface of heat structure index C. (W/m^2-K)
HS-HTC-POOL-L.C	Pool heat transfer coefficient at left (inside) boundary surface of heat structure index C. (W/m^2-K)
HS-HTC-POOL-R.C	Pool heat transfer coefficient at right (outside) boundary surface of heat structure index C. (W/m^2-K)
HS-ITER-FREQ	Average iteration frequency over all heat structures.
HS-MASS-FLUX-L.C	Mass flux to left (inside) boundary surface of heat structure index C. (kg/m^2-s)

HS-MASS-FLUX-R.C	Mass flux to right (outside) boundary surface of heat structure index C. (kg/m^2-s)
HS-MTC-L.C	Diffusion mass transfer coefficient at left (inside) boundary surface of heat structure index C. (m/s)
HS-MTC-R.C	Diffusion mass transfer coefficient at right (outside) boundary surface of heat structure index C. (m/s)
HS-POOL-FRAC-L.C	Fraction of left (inside) boundary surface of heat structure index C in pool of its boundary volume.
HS-POOL-FRAC-R.C	Fraction of right (outside) boundary surface of heat structure C in pool of its boundary volume.
HS-QFLUX-ATMS-L.C	Heat flux to atmosphere at left (inside) boundary surface of heat structure index C. (W/m^2)
HS-QFLUX-ATMS-R.C	Heat flux to atmosphere at right (outside) boundary surface of heat structure index C. (W/m^2)
HS-QFLUX-POOL-L.C	Heat flux to pool at left (inside) boundary surface of heat structure index C. (W/m^2)
HS-QFLUX-POOL-R.C	Heat flux to pool at right (outside) boundary surface of heat structure index C. (W/m^2)
HS-QTOTAL-ATMS-L.C	Time and surface integral of heat flux to atmosphere at left (inside) boundary surface of heat structure index C. (J)

HS-QTOTAL-ATMS-R.C	Time and surface integral of heat flux to atmosphere at right (outside) boundary surface of heat structure index C. (J)
HS-QTOTAL-POOL-L.C	Time and surface integral of heat flux to pool at left (inside) boundary surface of heat structure index C. (J)
HS-QTOTAL-POOL-R.C	Time and surface integral of heat flux to pool at right (outside) boundary surface of heat structure index C. (J)
HS-TEMP.CMN	Temperature at node MN of the heat structure index C. (K)

5 Control Function Arguments

The control function arguments are described below in this section. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

HS-CPUC	Total CPU time used for HS package calculations. (s)
HS-CPUE	Total CPU time used for HS package edits. (s)
HS-CPUR	Total CPU time used for HS package restarts. (s)
HS-DEGAS-ENERGY(NameSRC)	Total energy of released gas from gas source NameSRC. (J)
HS-DEGAS-MASS(NameSRC)	Total mass of released gas from gas source NameSRC. (kg)
HS-DEGAS-RATE(NameSRC)	Degassing rate of gas source NameSRC. (kg/s)

HS-DEGAS-STEELM(NameHS)	Steel mass melted during the system cycle for the steel degassing source associated with the heat structure NameHS. (kg)
HS-DEGAS-CSTEELM(NameHS)	Cumulative mass of steel melted for steel degassing source associated with the heat structure NameHS. (kg)

In the following variables acceptable strings for boundary surface sn are

LHS - Left (inside) boundary surface
RHS - Right (outside) boundary surface

HS-DELE-ATMS(NameHS,sn)	Integrated energy transfer to boundary volume atmosphere of boundary surface sn of heat structure NameHS due to mass transfer. (J)
HS-DELE-POOL(NameHS,sn)	Integrated energy transfer to boundary volume pool of boundary surface sn of heat structure NameHS due to mass transfer. (J)
HS-DELE-SURF(NameHS,sn)]	Integrated energy transfer to boundary volume surface of boundary surface sn of the heat structure NameHS due to mass transfer. (J)
HS-DELM-DROP(NameHS,sn)	Integrated droplet (fog) mass transfer to boundary volume atmosphere of boundary surface sn of heat structure NameHS. (kg)
HS-DELM-POOL(NameHS,sn)	Integrated water mass transfer to boundary volume pool of boundary surface sn of heat structure NameHS. (kg)
HS-DELM-STEAM(NameHS,sn)	Integrated vapor (steam) mass transfer to boundary volume atmosphere of boundary surface sn of heat structure NameHS. (kg)

HS-ENERGY-FLUX(NameHS,sn)	Energy flux associated with the corresponding HS-MASS-FLUX(NameHS,sn). (W/m2)
HS-ENERGY-INPUT(NameHS)	Energy input to heat structure NameHS by internal and surface power sources and by other packages. (J)
HS-ENERGY-STORED(NameHS)	Stored energy of heat structure NameHS. (J)
HS-FILM-ENTH(NameHS,sn)	Specific enthalpy of liquid film on boundary surface sn of heat structure NameHS. (J/kg)
HS-FILM-TEMP(NameHS,sn)	Temperature of film/atmosphere interface at boundary surface sn of heat structure NameHS. (K)
HS-FILM-MASS(NameHS,sn)	Mass of liquid film on boundary surface sn of heat structure NameHS. (kg)
HS-FILM-THICK(NameHS,sn)	Thickness of liquid film on boundary surface sn of heat structure NameHS. (m)
HS-HTC-ATMS(NameHS,sn)	Atmosphere heat transfer coefficient at boundary sn surface of heat structure NameHS. (W/m2-K)
HS-HTC-POOL(NameHS,sn)	Pool heat transfer coefficient at boundary surface sn of heat structure NameHS. (W/m2-K)
HS-ITER-FREQ	Average iteration frequency over all heat structures.
HS-MASS-FLUX(NameHS,sn)	Mass flux to boundary surface sn of heat structure NameHS. (kg/m2-s)

HS-MTC(NameHS,sn)	Diffusion mass transfer coefficient at boundary surface sn of heat structure NameHS. (m/s)
HS-POOL-FRAC(NameHS,sn)	Fraction of boundary surface sn of heat structure NameHS in the pool of its boundary volume.
HS-QFLUX-ATMS(NameHS,sn)	Heat flux to atmosphere at boundary surface sn of heat structure NameHS. (W/m2)
HS-QFLUX-POOL(NameHS,sn)	Heat flux to pool at boundary surface sn of heat structure NameHS. (W/m2)
HS-QTOTAL-ATMS(NameHS,sn)	Time and surface integral of heat flux to atmosphere at boundary surface sn of heat structure NameHS. (J)
HS-QTOTAL-POOL(NameHS,sn)	Time and surface integral of heat flux to pool at boundary sn surface of heat structure NameHS. (J)
HS-RAD-FLUX(NameHS,sn)	Total structure-to-structure radiation to radiative heat flux from boundary surface sn of heat structure NameHS. (J)
HS-RE-POOL(NameHS, sn)	Reynolds number for pool at boundary sn surface of heat structure NameHS. (-)
HS-RE-ATM(NameHS, sn)	Reynolds number for atmosphere at boundary sn surface of heat structure NameHS. (-)
HS-PR-POOL(NameHS, sn)	Prandtl number for pool at boundary sn surface of heat structure NameHS. (-)
HS-PR-ATM(NameHS, sn)	Prandtl number for atmosphere at boundary sn surface of heat structure NameHS. (-)

HS-PRS-POOL(NameHS, sn)	Wall Prandtl number for pool at boundary sn surface of heat structure NameHS. (-)
HS-PRS-ATM(NameHS, sn)	Wall Prandtl number for atmosphere at boundary sn surface of heat structure NameHS. (-)
HS-NU-POOL(NameHS, sn)	Nusselt number for pool at boundary sn surface of heat structure NameHS. (-)
HS-NU-ATM(NameHS, sn)	Nusselt number for atmosphere at boundary sn surface of heat structure NameHS. (-)
HS-TEMP(NameHS,MN)	Temperature at node MN of heat structure NameHS. (K)

6 Example Input

This section provides example input for a problem that uses the HS package for MELGEN and MELCOR execution. This problem is included to provide a non-trivial example of the input for the HS package and is not intended to be a reactor safety application of the MELCOR Code.

Consider a boiling water reactor (BWR) with a Mark II containment. The reactor produces 3226 MW of thermal power from 47,368 nuclear fuel rods. The example problem whose input is discussed here is a boil-off with the core initially 85% covered. The following packages are used in this calculation:

(1) HS package

(2) CVH package

(3) FL package

(4) MP package

(5) CF package

(6) TF package

This example problem does not use the COR package. The core is represented by a single fuel rod heat structure with multiplicity 47,368. The following sections contain illustrations of input for this problem.

6.1 Example MELGEN Input

A subset of the MELGEN input that will create a restart file to execute the boil-off problem is given here. This input contains records not only for heat structures, but also for some concomitant gas source, noncondensible gas, material properties, and tabular function input.

6.1.1 Heat Structure Input

The following illustrates complete input records for 3 of 21 heat structures in the example problem. These provide examples of various nodalizations, material compositions, geometries, power sources, and boundary conditions for heat structures. See Section 2.1.1 for details on MELGEN heat structure inputs.

The following records are comments (records beginning with the symbol !) and a complete set of input for an intact fuel rod in the LaSalle nuclear reactor. This heat structure is a vertical structure with a cylindrical geometry. It has a multiplicity equal to the number of fuel rods in the core. There are 12 temperature nodes. Each fuel rod is composed of uranium dioxide fuel, a helium gap, and a Zircaloy cladding. The decay-heat power after scram is represented by an internal power source whose power is given as a function of time by a tabular function. A symmetry boundary condition is applied at the center, and a convective boundary condition for external flow with calculated heat transfer coefficient is applied at the surface.

```
! INTACT FUEL RODS (5.4943-9.3043 M) – HEAT TRANSFER TO CORE
HS_ID 'FUEL ROD'
HS_GD CYLINDRICAL      YES
HS_EOD   5.4943     1.0
HS_MLT   4.7368E4
HS_ND   12 !N  NXVALU XVALUE       TEMPIN    MATNAM       QFRCIN
            1    1     0.0          318.0 uranium-dioxide   1.0
            2    2     8.678160E-4  318.0 uranium-dioxide   1.0
            3    3     1.735632E-3  318.0 uranium-dioxide   1.0
            4    4     2.603448E-3  318.0 uranium-dioxide   1.0
            5    5     3.471264E-3  318.0 uranium-dioxide   1.0
            6    6     4.339080E-3  318.0 uranium-dioxide   1.0
            7    7     5.206896E-3  318.0 helium           0.0
            8    8     5.321196E-3  318.0 zircaloy          0.0
            9    9     5.524421E-3  318.0 zircaloy          0.0
            10   10    5.727646E-3  318.0 zircaloy          0.0
            11   11    5.930871E-3  318.0 zircaloy          0.0
            12   12    6.134096E-3  318.0
HS_SRC   TF   'DECAY-HEAT POWER'   2.0889E-5
HS_LB   Symmetry
HS_RB   CalcCoefHS   CV101   YES
HS_RBP   EXT   0.5   0.5
HS_RBR   .5   EQUIV-BAND   .1
HS_RBS   1.429449E-1 1.226819E-2   3.70884
```

The following records are comments and a complete set of input for the upper head of the pressure vessel for the reactor. The upper head has a hemispherical geometry and

is made of carbon steel. A convective boundary condition for internal flow with calculated heat transfer coefficients is applied at the interior surface of the reactor, and a convective boundary condition with a heat transfer coefficient obtained from a tabular function is applied at the exterior surface. The exterior surface boundary condition represents the insulation on the pressure vessel.

```
! UPPER HEAD – UPPER PLENIUM TO DRYWELL HEAT TRANSFER
!
HS_ID   'UPPER HEAD'
HS_GD   TOPHALFSPHERE   NO
HS_ND   5   2!   N  NXVALU XVALUE    TEMPIN   MATNAM
                 1    1     3.226    560.0  CARBON-STEEL
                 2    5     3.334    560.0
HS_EOD   18.53
HS_SRC   NO
HS_LB   CalcCoefHS   CV104   YES
HS_LBP   INT   0.5   0.5
HS_LBR   .75   EQUIV-BAND   3.
HS_LBS   50.59   3.226   3.226
HS_RB   CoefTempTF   'HTC OUT VESS'   CV205   YES
HS_RBP   EXT   0.5   0.5
HS_RBR   .3   EQUIV-BAND   15.
HS_RBS   54.55   3.334   3.334
```

The following records are comments and a complete set of input for the wetwell floor of the containment for the nuclear power plant. It is represented as a rectangular, horizontal heat structure made of concrete. At both surfaces, a convective boundary condition with calculated heat transfer coefficient is applied.

```
! WETWELL FLOOR
! WETWELL TO SECONDARY CONTAINMENT HEAT TRANSFER
HS_ID   'WETWELL FLOOR'
HS_GD   RECTANGULAR   NO
HS_ND   5   2!   N  NXVALU XVALUE    TEMPIN   MATNAM
                 1    1     0.0      325.0   CONCRETE
                 2    5     3.048    325.0
HS_EOD   -28.7274   0.0
HS_SRC   NO
HS_LB   CalcCoefHS   CV300   YES
HS_LBP   EXT   0.5   0.5
HS_LBR   .8   EQUIV-BAND   15.
HS_LBS   548.0   26.416   26.416
HS_RB   CalcCoefHS   CV200   YES
HS_RBP   'EXT'   0.5   0.5
HS_RBR   .8   EQUIV-BAND   18.
HS_RBS   548.0   26.416   26.416
```

6.1.2 Gas Source Input

The following records are a complete set for a gas source that models the degassing of limestone concrete by the release of free water. See Section 2.1.2 for details on MELGEN gas source input.

The source releases water vapor into the boundary volume of the left boundary surface of heat structure 20001 between the temperatures 360 K and 380 K. It is distributed over the entire heat structure and can release as much as 2.211×10^5 kg of water vapor.

```
HS_DG    SRC1    HS20001    GAS    LeftBnd    4    H2O-VAP
HS_DGSRC     110.0      1.808E6    360.0       380.0
```

Limestone concrete can also release water vapor from bound water and carbon dioxide from calcium carbonate as its temperature increases. The following records are input for the HS package degassing model for gas sources for the degassing of the concrete in the heat structure named HS20001:

```
HS_DG     SRC2    HS20001    GAS    LeftBnd    4    CO2
HS_DGSRC    72.0    6.912E6    500.0    520.0
HS_DG     SRC3    HS20001    GAS    LeftBnd    4    H2O-VAP
HS_DGSRC    110.0    2.909E7    360.0    380.0
```

6.1.3 Noncondensible Gas (NCG) Input

The following records specify the noncondensible gases for the example problem. The carbon dioxide specification must be present since it is released by a gas source for the degassing model. The water vapor that is released by some gas sources is a material that is present in all problems. See the NCG Package Users' Guide for details on NCG MELGEN input.

```
NCG_INPUT
NCG_ID H2        ! HYDROGEN
NCG_ID N2        ! NITROGEN
NCG_ID CO2       ! CARBON DIOXIDE
```

6.1.4 Material Property (MP) Input

The following records illustrate the material properties input that is required for the new material 'CONCRETE'. For this problem, input for the new materials carbon steel and helium must also be present. Tabular functions TF106, TF107, TF108, TF200, TF201, TF202 must also be provided for the thermal conductivity, heat capacity, and density of concrete and helium. Refer to the MP Package Users' Guide for details on MP MELGEN input.

```
MP_INPUT
MP_ID    CONCRETE
MP_PRTF 3 ! NPAR PROPERTY DEFAULT/TF/CF(may be only for THC)
             1    THC      TF106            TF
             2    CPS      TF107
             3    RHO      TF108
MP_ID    helium
MP_PRTF 3 ! NPAR PROPERTY DEFAULT/TF/CF(may be only for THC)
             1    THC      TF200      TF
             2    CPS      TF201
             3    RHO      TF202
```

6.1.5 Tabular Function (TF) Input

The following records are input for tabular function 'HTC OUT VESS', which is the heat transfer coefficient as a function of temperature for the insulated exterior of the reactor pressure vessel. This function specifies a constant heat transfer coefficient of 10 W/m^2-K. See the TF Package Users' Guide for details on TF MELGEN input.

```
TF_INPUT
! *************************
TF_ID    HTC OUT VESS    1.0    0.0
```

```
TF_TAB   2 ! NTFPAR   X        Y
               1   273.15    10.0
               2   5000.0    10.0
```

The following records are input for tabular function 'DECAY-HEAT POWER', which is the decay-heat power as a function of time for the LaSalle reactor core. This function specifies an initial power of 3326 MW with a decay that is typical of a BWR core after scram.

```
TF_ID    'DECAY-HEAT POWER'    3.326E9    0.0
TF_TAB   22 ! NTFPAR   X        Y
               1     0.0   1.000000
               2     0.1   0.900689
               3     0.2   0.274300
               4     0.3   0.153171
               5     0.4   0.110821
               6     0.5   0.091625
               7     0.6   0.083212
               8     0.8   0.073556
               9     1.0   0.064777
               10    1.5   0.063089
               11    2.0   0.059854
               12    3.0   0.057265
               13    4.0   0.055204
               14    6.0   0.052085
               15    8.0   0.049776
               16    10.0  0.047947
               17    15.0  0.044575
               18    20.0  0.042176
               19    30.0  0.038783
               20    40.0  0.036348
               21    60.0  0.031546
               22    1.0E5 0.001460
```

6.2 Example MELGEN Input for Ice Condenser Model

An example of heat structure, gas source, and tabular value input that was used to model a PWR ice condenser is provided below. See the descriptions related to input records HS_DG and HS_RB for details.

```
HS_INPUT
! ******************************
HS_ID   'Ice Condenser' ! name of structure
HS_GD    CYLINDRICAL      NO
HS_EOD   10.0   1.0 ! elevation is vertical
HS_MLT   1000.   ! multiplicity
HS_ND    2 !N  NXVALU XVALUE       TEMPIN    MATNAM
            1    1     0.0          274.    basket
            2    2     0.15         274.
HS_SRC   NO
HS_LB   Symmetry   ! adiabatic left surface
HS_RB   CalcCoefHS   CV1   YES ! convective H.T.C.
HS_RBP   'EXT'   0.5   0.5
HS_RBS   1.0   0.3   14.60 ! CLNR is diameter
! ******************************
HS_DG    SRC1    'Ice Condenser'    ICE    RightBnd    1    POOL
HS_DGSRC   1000. 754419. 274. 373.
HS_DGICE   1.2  0.33  1.
! ******************************
!
```

```
! basket properties
!
MP_INPUT
MP_ID    basket
MP_PRTF 3 ! NPAR PROPERTY DEFAULT/TF/CF(may be only for THC)
             1    RHO      'basket rho'
             2    CPS      'basket cps'
             3    THC      'basket thc'            TF
...
! ******************************
TF_INPUT
! *************************
TF_ID    'basket rho'   1.0
TF_TAB   1 ! NTFPAR  X    Y
                1   274. 1.
! *************************
TF_ID 'basket cps'    1.0
TF_TAB   1 ! NTFPAR  X    Y
                1   274. 485.7
! *************************
TF_ID 'basket thc'    1.0
TF_TAB   1 ! NTFPAR  X    Y
                1   274. 5.
```

6.3 Example MELGEN and MELCOR Input for Film-Tracking Model

An example of heat structure, film-tracking, tabular function, spray, and transfer process input that was used to model a passive containment cooling system for a hypothetical containment dome is provided below. See sections 2.1.3 and 2.2.3 for details on film-tracking input.

The MELGEN input for the film-tracking model is as follows:

```
CVH_INPUT
CV_ID 'containment' 'CVTYPE07'
CV_THR NonEquil FOG ACTIVE
CV_PAS SEPARATE PoolAndAtm SATURATED SUPERHEATED
CV_PTD PVOL   1.0e5
CV_AAD TATM   348.
CV_NCG   1 RHUM  1.0   !N  NAMGAS  MLFR
                        1   H2     1.0
CV_VAT   3   !N  CVZ     CVVOL
              1  0.0    0.0
              2  6.0    169.646
              3  9.0    226.195
CV_SOU   1  !N SourceInfo
             1 AE RATE TF 'enthalpy source'
...
! ****************************
CV_ID 'environment' 'CVTYPE06'
CV_THR NonEquil FOG ACTIVE
CV_PAS SEPARATE PoolAndAtm SATURATED SUPERHEATED
CV_PTD PVOL   1.0e5
CV_AAD TATM   298.
CV_NCG   1 RHUM  0.5   !N  NAMGAS  MLFR
                        1   H2     1.0
CV_VAT   2   !N  CVZ     CVVOL
              1  0.0     0.0
              2  48.0    1.e10
...
HS_INPUT
```

```
! ********************************
HS_ID   'bot. cyl.'            !  cyl. bottom section
HS_GD    CYLINDRICAL      YES
HS_EOD   0.0  1.0              !  elevation, vertical
HS_MLT   1.                   !  multiplicity
HS_ND    2 !N  NXVALU XVALUE       TEMPIN        MATNAM
            1    1     3.          274.    'stainless-steel'
            2    2     3.05         274.
HS_SRC   NO
HS_LB   CalcCoefHS   'containment'   YES !  convective h.t.c.
HS_LBP   int   0.5   0.5       !  convective h.t.c
HS_LBS   1. 3.e-0    3.0      !  clnl is radius
HS_RB   CalcCoefHS   'environment'   YES ! convective H.T.C.
HS_RBP   ext    0.5   0.5
HS_RBS   1.  3.e-0    3.0     !  clnr is radius
HS_FT    ON
HS_FTDRN    0
HS_FTLBF    0.0
HS_FTRBF    0.0
! ******************************
HS_ID   'mid. cyl.'    !  cyl. middle section
HS_GD    CYLINDRICAL      YES
HS_EOD   3.0  1.0              !  elevation, vertical
HS_MLT   1.                   !  multiplicity
HS_ND    2 !N  NXVALU XVALUE       TEMPIN        MATNAM
            1    1     3.          274.    'stainless-steel'
            2    2     3.05         274.
HS_SRC   NO
HS_LB   CalcCoefHS   'containment'   YES !  convective h.t.c.
HS_LBP   int   0.5   0.5       !  convective h.t.c
HS_LBS   1.  3.e-0   3.0      !  clnl is radius
HS_RB   CalcCoefHS   'environment'   YES ! convective H.T.C.
HS_RBP   ext 0.5   0.5
HS_RBS   1.  3.e-0   3.0      !  clnr is radius
HS_FT   ON
HS_FTDRN 1 !N   IDSDRN      FDRNL   FDRNR
            1  'bot. cyl.'   1.0     1.0
HS_FTLBF   0.0
HS_FTRBF   0.0
! ******************************
HS_ID   'top hemi.'            !  hemispherical dome HS300
HS_GD    TOPHALFSPHERE      YES
HS_EOD   6.0  1.0              !  elevation, vert. (unused)
HS_MLT   1.                   !  multiplicity
HS_ND    2 !N  NXVALU XVALUE       TEMPIN        MATNAM
            1    1     3.          274.    'stainless-steel'
            2    2     3.05         274.
HS_SRC   NO
HS_LB   CalcCoefHS   'containment'   YES !  convective h.t.c.
HS_LBP   int   0.5   0.5       !  convective h.t.c
HS_LBS   1.  3.e-0 3.0        !  clnl is radius
HS_RB   CalcCoefHS   'environment'   YES ! convective H.T.C.
HS_RBP   ext    0.5   0.5
HS_RBS   1.  3.e-0    3.0     !  clnr is radius
HS_FT   ON
HS_FTDRN 1 !N   IDSDRN      FDRNL   FDRNR
            1 'mid. cyl.'    0.5     1.0
HS_FTLBF     0.5 TP100
HS_FTRBF     0.0
HS_FTRBM   TF   'water source'
HS_FTRBE   TF   'water enthalpy'
! ***********************************
TF_INPUT
```

```
!
! *************************
TF_ID    'water source'   1.0 ! passive containment cooling
TF_TAB   3 ! NTFPAR   X     Y
               1   0.0    3.e0
               2   1.5e3  3.e0
               3   4.e3   7.0
! *************************
TF_ID    'water enthalpy'   1.0
TF_TAB   2 ! NTFPAR   X     Y !  water at about 300 K
               1   0.0    1.e5
               2   1.e4   1.e5
! *************************
TF_ID    'enthalpy source'   1.0
TF_TAB   4 ! NTFPAR   X     Y ! simulates containment heating
               1   0.0  0.0
               2   10. 0.0
               3   10.1  1.0e7
               4   1.e4 1.0e7
! *************************
SPR_INPUT
SPR_ID   'rain'   'containment'   8. !  rain from HS 'top hemi.'
SPR_DTFR    TP   TP100  !  "out" Trans. Pr. TP100
SPR_DSD   1 ! NT     DIAMO   DRFREQ
            1      3.e-3     1. ! drop size (3 mm)
! *************************
TP_INPUT
TP_IN   1 ! NAMETPIN *** NAMEPAC DIRECT IP2EDF ** NMSIN NHTRM **
          1  TP100            HS
TP_OUT  1 ! NAMETPOT *** NPOTOI *** OUTMTX ******** NMSOT**
          1  TP100       TP100       DEF1            1
The MELCOR input (to change the drainage partitioning on the left surface of structure
'top hemi.') is as follows:
HS_ID 'top hemi.'
HS_FT ON
HS_FTDRN 1 !N   IDSDRN      FDRNL   FDRNR
            1 'mid. cyl.'    0.8    1.0
HS_FTLBF    0.2 TP100
HS_FTRBF     0.0
```

7 Example Output

Both time-independent and time-dependent data from the database for the HS package are printed during MELGEN and MELCOR execution. Sections 7.1 and 7.2 discuss the printed output from these data, respectively. Additional information is printed during MELGEN and MELCOR execution. See sections 7.3 and 7.4 for details on the output containing this information.

7.1 Time-independent Data Output

During MELGEN execution after all input data are processed and all initialization calculations are completed, and during MELCOR execution prior to commencing the first cycle of a restart, time-independent data are printed. For the HS package, this output follows the header:

EDIT FOR HEAT STRUCTURE PACKAGE
(TIME-INDEPENDENT DATA)

For the heat structure with user-input name HSNAME, the following information is printed in the section "HEAT STRUCTURE HSNAME":

(1) Number of temperature nodes

(2) Location of temperature nodes

(3) Geometry type

(4) Altitude of base

(5) Orientation parameter

(6) Multiplicity

(7) Name of material in each mesh interval

(8) Mass in each mesh interval

(9) Total mass of structure

(10) Volume in each mesh interval

(11) Total volume of structure

(12) Internal power source information (tabular function name for power table if source exists or statement that no internal power source exists)

(13) Power fractions for each mesh interval if an internal power source exists, or statement that all power fractions are zero if no internal power source exists

(14) Boundary volume name for each boundary surface

(15) Type of boundary condition which is applied at each boundary surface

(16) Type of flow (internal/external) over each boundary surface

(17) Surface power source information for each boundary surface (tabular function name for power table if source exists, or statement that no surface power source exists)

(18) Statement indicating whether or not mass transfer calculations are performed for each boundary surface

(19) Area of each boundary surface

(20) Characteristic length of each boundary surface

(21) Axial length of each boundary surface

(22) Critical pool fractions for pool and atmosphere heat transfer.

At each boundary surface whose critical pool fraction for pool heat transfer exceeds its fraction for atmosphere heat transfer, heat transfer to neither the pool nor the atmosphere will be calculated for pool fractions between these values. If at least one surface has such a specification, information is printed for each such surface in the section that follows the statement

> WARNING – HEAT TRANSFER MAY NOT BE CALCULATED AT THE FOLLOWING SURFACES:

This information includes the following:

(1) Identification name of heat structure.

(2) Side of the heat structure—left (inside) or right (outside)—for which no heat transfer might be calculated

(3) Range of pool fraction for which no heat transfer will be calculated.

If no gas sources for the HS package degassing model are present, the statement

> NO GAS SOURCES FOR THE DEGASSING MODEL ARE PRESENT IN THIS PROBLEM

is printed. If the gas source with user input-name GSNAME exists for a problem, the following information is printed in the section "GAS SOURCE GSNAME":

(1) Mesh intervals over which the source is distributed

(2) Name of gas that is released and number of the control volume into which it is released

(3) Density

(4) Heat of reaction

(5) Boundaries of the temperature range over which the gas is released

(6) Total mass of the gas that can be released The following is an example of the time-independent output:

```
                                        ****************************************
                                        *     EDIT FOR HEAT STRUCTURE PACKAGE    *
                                        *         (TIME-INDEPENDENT DATA)        *
                                        ****************************************

*****************************************************************************************************************

                                          HEAT STRUCTURE HS10001

 NUMBER OF TEMPERATURE NODES: 05
 LOCATION OF TEMPERATURE NODES (M):
      01/ 2.200E+00 02/ 2.205E+00 03/ 2.277E+00 04/ 2.348E+00 05/ 2.420E+00
 GEOMETRY TYPE: CYLINDRICAL
 ALTITUDE OF BASE (M):  5.400E+00
 ORIENTATION PARAMETER:  1.000E+00
 MULTIPLICITY:  1.000E+00
 MATERIAL IN EACH MESH INTERVAL:
      01/ STAINLESS-STEEL            02/ CARBON-STEEL               03/ CARBON-STEEL               04/ CARBON-STEEL
 MASS IN EACH MESH INTERVAL  (KG):
      01/ 4.170E+03 02/ 5.945E+04 03/ 6.136E+04 04/ 6.326E+04
 TOTAL MASS  (KG):  1.882E+05
 VOLUME IN EACH MESH INTERVAL  (M**3):
      01/ 5.259E-01 02/ 7.669E+00 03/ 7.914E+00 04/ 8.159E+00
 TOTAL VOLUME  (M**3):  2.427E+01
 NO INTERNAL POWER SOURCE (ALL POWER FRACTIONS ARE ZERO)

                                LEFT (INSIDE) BOUNDARY SURFACE                        RIGHT (OUTSIDE) BOUNDARY SURFACE
 BOUNDARY VOLUME                                CV160                                                  CV300
 BOUNDARY CONDITION    CONVECTIVE WITH CALCULATED HEAT TRANSFER COEFFICIENTS   CONVECTIVE WITH CALCULATED HEAT TRANSFER COEFFICIENTS
                                          INTERNAL FLOW                                          EXTERNAL FLOW
                                     EMISSIVITY =    0.00000                                EMISSIVITY =    0.00000
                                     NO SURFACE POWER SOURCE                                NO SURFACE POWER SOURCE
                                    MASS TRANSFER CALCULATED                               MASS TRANSFER CALCULATED
 SURFACE AREA (M**2)                        1.051E+02                                              1.156E+02
 CHARACTERISTIC LENGTH (M)                  1.000E+00                                              1.250E+01
 AXIAL LENGTH (M)                           7.600E+00                                              7.600E+00
 MULTIPLIERS FOR CALCULATED COEFFICIENTS
      HEAT TRANSFER                         1.000E+00                                              1.000E+00
      MASS TRANSFER                         1.000E+00                                              1.000E+00
 CRITICAL POOL FRACTIONS
      POOL                                  5.000E-01                                              5.000E-01
      ATMOSPHERE                            5.000E-01                                              5.000E-01
 CRITICAL TEMP DIFFERENCES
      POOL                                  1.000E+02                                              1.000E+02
      ATMOSPHERE                            1.000E+02                                              1.000E+02
```

7.2 Time-Dependent Data Output

During MELGEN execution after all input data are processed and all initialization calculations are completed and during MELCOR execution at user-defined time intervals, time-dependent data are printed. For the HS package, this output follows the header

```
EDIT FOR HEAT STRUCTURE PACKAGE
(TIME-DEPENDENT DATA)
TIME = 0.000E+00 CYCLE = 000000000.
```

The version number and creation date of the HS package and its associated manuals are printed at the top of the page containing this header. The units for each output quantity are given following this header.

The output for the heat structures is printed in the following four sections:

(1) HEAT STRUCTURE TEMPERATURE DISTRIBUTIONS

(2) HEAT TRANSFER DATA

(3) MASS TRANSFER DATA

(4) ENERGY TRANSFER DATA.

The temperature at each node in each heat structure is printed in the section “HEAT STRUCTURE TEMPERATURE DISTRIBUTIONS”. The temperatures for the heat structure with user-defined name HSNAME are printed on lines following a left-justified HEAT STRUCTURE HSNAME.

The following data for the boundary surfaces of each heat structure are printed in the section HEAT TRANSFER DATA:

(1) Boundary volume

(2) Surface temperature

(3) Pool fraction (fraction of surface in the pool of its boundary volume)

(4) Atmosphere convective heat transfer coefficient

(5) Atmosphere radiation heat transfer coefficient

(6) Pool heat transfer coefficient

(7) Atmosphere heat transfer flow regime

(8) Pool heat transfer flow regime.

The following data for each boundary surface of each heat structure are printed in the section MASS TRANSFER DATA:

(1) Boundary volume

(2) Pool fraction

(3) Mass flux

(4) Liquid film thickness

(5) Liquid film mass

(6) Changes in the mass of steam, fog droplets, and the pool in the boundary volume due to mass transfer at the surface.

The following data for each heat structure are printed in the section ENERGY TRANSFER DATA:

(1) Stored energy

(2) Energy input by internal and surface power sources and by transfers from other packages.

The following data are printed in this section for each boundary surface:

(1) Specific enthalpy of its liquid film

(2) Time and surface integral of its atmosphere and pool heat fluxes

(3) Energy transferred to the atmosphere and the pool of its boundary volume due to mass transfer.

The information printed in this section at different times is sufficient to establish the global conservation of energy for the HS package.

If gas sources are present in the problem, the following data are printed for each gas source in the section "DEGASSING SOURCE DATA":

(1) Name of heat structure that contains the source

(2) Control volume into which gas is released

(3) Degassing rate

(4) Total mass of gas released

(5) Total energy consumed by gas release.

7.3 Additional MELGEN Output

Prior to the time-independent data edit, the following information is given for the HS package during MELGEN execution:

(1) Diagnostic messages if input errors are detected during processing of input records that are internal to the HS package (pass-one processing)

(2) Total number of heat structures and gas sources

(3) CPU time that was required for pass-one input processing

(4) Diagnostic messages if input errors are detected that result from inconsistencies with other packages (pass-two processing).

7.4 Additional MELCOR Output

Additional MELCOR output provided by the HS package is discussed in this section.

Each edit includes the section HEAT STRUCTURE PACKAGE STATISTICS, which documents the performance of the HS package. For the entire problem and for the portion of the problem between the past and present edits, the following information is printed:

(1) CPU time for calculation, edit, and restart

(2) Average iteration frequency per heat structure.

8 Diagnostics and Error Messages

The HS package prints a message if any of the following occur:

(1) An error is detected during MELGEN input processing.

(2) Temperature convergence during steady-state initialization in MELGEN execution is not achieved.

(3) An error is detected during MELCOR input processing.

(4) During MELCOR execution, the HS package requests repeating a computational cycle with a reduced timestep.

See sections 8.1 through 8.3 for details on the messages discussed.

8.1 MELGEN input Errors

Error messages are printed by the HS package when errors are detected during input processing in MELGEN execution. Such errors occur if input specifications are violated or if input data are inconsistent. In general, any input that violates a “must” imperative in

the user-input descriptions of Sections 2.1.1 and 2.1.2 will be detected as an error. All such error messages are self-explanatory and indicate which input record must be corrected. Error messages are also printed if materials, control volumes, or tabular functions, which are specified in heat structure input, are not present in a problem. Concomitant with the detection of an error during MELGEN input processing is a directive to not create a restart file.

Failure to converge in a steady-state initialization temperature iteration usually indicates a problem associated with the user inputs related to the heat structure nodalization or CVH boundary volume. In this case, the user should re-examine the input for errors or unrealistic conditions. As a last resort, the user can abandon steady-state initialization for that heat structure and prescribe a temperature profile with the HS_ND tabular record.

8.2 MELCOR Input Errors

Error messages are printed by the HS package when errors are detected during input processing in MELCOR execution. All such error messages are self-explanatory and indicate which input record must be corrected. Error messages are also printed if materials, control volumes, or tabular functions, which are specified in heat structure input, are not present in a problem. Concomitant with any input error message during MELCOR execution is a directive to not proceed with a transient calculation.

8.3 Requests to Repeat a Computational Cycle

The HS package prints a diagnostic message during MELCOR execution whenever it requests repeating a computational cycle with a reduced timestep because of numerical difficulties. MELCOR only terminates execution after a timestep reduction if the new timestep is less than the minimum specified by the user. In such a situation, a restart file is written, and an edit of all current data is performed.

Appendix 1

Melcor 2.1	PTFRead	Comments
HS-CPUC	HS-CPUC	Total CPU time used for HS package calculations
HS-CPUE	HS-CPUE	Total CPU time used for HS package edits
HS-CPUR	HS-CPUR	Total CPU time used for HS package restarts
HS-DEGAS-ENERGY (NameSRC) NameSRC – the gas source name	HS-DEGAS-ENERGY.G G – the gas source index	Total energy of released gas from gas source NameSRC (G)
HS-DEGAS-MASS (NameSRC) NameSRC – the gas source name	HS-DEGAS-MASS. G G – the gas source index	Total mass of released gas from gas source NameSRC (G)
HS-DEGAS-RATE (NameSRC) NameSRC – the gas source name	HS-DEGAS-RATE. G G – the gas source index	Degassing rate of gas source NameSRC (G)
HS-DEGAS-STEELM (NameHS) NameHS – the heat structure name	HS-DEGAS-STEELM. G G – the gas source index	Steel mass melted during the system cycle for steel degassing source index G (associated with heat structure NameHS)
HS-DEGAS-CSTEELM (NameHS) NameHS – the heat structure name	HS-DEGAS-CSTEELM. G G – the gas source index	Cumulative mass of steel melted for steel degassing source associated index G (associated with heat structure NameHS)
HS-DELE-ATMS (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-DELE-ATMS-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Integrated energy transfer to boundary volume atmosphere of boundary surface sn (S) of heat structure NameHS (C) due to mass transfer
HS-DELE-POOL (NameHS,sn)	HS-DELE-POOL-S.C	Integrated energy transfer to boundary volume pool of

Melcor 2.1	PTFRead	Comments
NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	boundary surface sn (S) of heat structure NameHS (C) due to mass transfer
HS-FILM-TEMP (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-FILM-TEMP-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Temperature of film/atmosphere interface at boundary surface sn (S) of heat structure NameHS (C)
HS-DELM-DROP (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-DELM-DROP-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Integrated droplet (fog) mass transfer to boundary volume atmosphere of boundary surface sn (S) of heat structure NameHS (C)
HS-DELM-POOL (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-DELM-POOL-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Integrated water mass transfer to boundary volume pool of boundary surface sn (S) of heat structure NameHS (C)
HS-DELM-STEAM (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary	HS-DELM-STEAM-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface	Integrated vapor (steam) mass transfer to boundary volume atmosphere of boundary surface sn (S) of heat structure NameHS (C)

Melcor 2.1	PTFRead	Comments
surface RHS - right (outside) boundary surface	R - right (outside) boundary surface	
HS-ENERGY-FLUX (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-ENERGY-FLUX-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Energy flux associated with HS-MASS-FLUX (NameHS,sn) (HS-MASS-FLUX-S.C)
HS-ENERGY-INPUT (NameHS) NameHS – the heat structure name	HS-ENERGY-INPUT.C C – the heat structure index	Energy input to heat structure NameHS (C) by internal and surface power sources and by other packages
HS-ENERGY-STORED (NameHS) NameHS – the heat structure name	HS-ENERGY-STORED. C C – the heat structure index	Stored energy of heat structure NameHS (C)
HS-FILM-ENTH (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-FILM-ENTH-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Specific enthalpy of liquid film on boundary surface sn (S) of heat structure NameHS (C)
HS-FILM-MASS (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-FILM-MASS-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Mass of liquid film on boundary surface sn (S) of heat structure NameHS (C)
HS-FILM-THICK (NameHS,sn)	HS-FILM-THICK-S.C	Thickness of liquid film on boundary surface sn (S) of

Melcor 2.1	PTFRead	Comments
NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	C – the heat structure index; S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	heat structure NameHS (C)
HS-HTC-ATMS(NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-HTC-ATMS-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Atmosphere heat transfer coefficient at boundary surface sn (S) of heat structure NameHS (C)
HS-HTC-POOL(NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-HTC-POOL-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Pool heat transfer coefficient at boundary surface sn (S) of heat structure NameHS (C)
HS-ITER-FREQ	HS-ITER-FREQ	Average iteration frequency over all heat structures
HS-MASS-FLUX (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-MASS-FLUX-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Mass flux to boundary surface sn (S) of heat structure NameHS (C)
HS-MTC(NameHS,sn) NameHS – the heat structure name sn – the boundary surface:	HS-MTC-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary	Diffusion mass transfer coefficient at boundary surface sn (S) of heat structure NameHS (C)

Melcor 2.1	PTFRead	Comments
LHS - left (inside) boundary surface RHS - right (outside) boundary surface	surface R - right (outside) boundary surface	
HS-POOL-FRAC (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-POOL-FRAC-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Fraction boundary surface sn (S) of heat structure NameHS (C) in pool of its boundary volume
HS-QFLUX-ATMS (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-QFLUX-ATMS-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Heat flux to atmosphere at boundary surface sn (S) of heat structure NameHS (C)
HS-QFLUX-POOL (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-QFLUX-POOL-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Heat flux to pool at boundary surface sn (S) of heat structure NameHS (C)
HS-QTOTAL-ATMS (NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-QTOTAL-ATMS-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Time and surface integral of heat flux to atmosphere at boundary surface sn (S) of heat structure NameHS (C)
HS-QTOTAL-POOL	HS-QTOTAL-POOL-S.C	Time and surface integral of

Melcor 2.1	PTFRead	Comments
(NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	heat flux to pool at boundary surface sn (S) of heat structure NameHS (C)
HS-RAD-FLUX(NameHS,sn) NameHS – the heat structure name sn – the boundary surface: LHS - left (inside) boundary surface RHS - right (outside) boundary surface	HS-RAD-FLUX-S.C C – the heat structure index S – the boundary surface: L - left (inside) boundary surface R - right (outside) boundary surface	Total structure-to-structure radiation to radiative heat flux from boundary surface sn (S) of heat structure NameHS (C)
HS-TEMP(NameHS,MN) NameHS – the heat structure name MN – the node number	HS-TEMP.CMN C – the heat structure index MN – the node number	Temperature at node MN of heat structure NameHS (C)

Material Properties (MP) Package
Users' Guide

The MELCOR Material Properties (MP) package models the physical properties needed by many of the various physics packages. This is done by using analytical laws, correlations, or linear tables. New materials and their properties may be defined through user input, and properties for default materials may be redefined by user input.

This Users' guide gives a list of the default materials and the properties defined in the package, describes the user input, and lists some sample input and output.

The default property values and functions used in the MP package along with their references are provided in the MP Package Reference Manual.

Contents

LIST OF TABLES **2**

1 1. INTRODUCTION **3**

1.1 Default Material Properties 3

1.2 Changes since Version 1.8.6 5

1.2.1 Adjustment of Melt Point for Interacting Materials 5

1.3 Changes for F95 Version 6

2 USER INPUT **6**

2.1 MELGEN Input 6

MP_ID – Material Name 6
MP_DELETE – Ignore all records associated with MP_ID 7
MP_PRTF – Property, Tabular Function Pairs 7
MP_PRC – Property, Constant Values 9
MP_PVE – Viscosity and thermal expansivity of oxide and metallic molten pools 9
MP_SS – Steel Composition (only for stainless steel [SS]) 10

2.2 MELCOR Input 11

3 SENSITIVITY COEFFICIENTS **11**

4 PLOT VARIABLES AND CONTROL FUNCTION ARGUMENTS **11**

5 EXAMPLE MELGEN INPUT **12**

6 EXAMPLE MELGEN OUTPUT (PARTIAL LISTING) **12**

7 DIAGNOSTIC AND ERROR MESSAGES **13**

List of Tables

Table 1.1 MP Materials and Mnemonic Identifiers 3
Table 1.2 Defined Material Properties (Mnemonic) 4
Table 1.3 Default Material Properties, Property Mnemonics, and User Input Capabilities 5
Table 2.1 Default Material Names 7

1 Introduction

1.1 Default Material Properties

The MELCOR Material Properties (MP) package models many common properties needed by the various phenomenological packages through the use of analytical laws, correlations, and tabulated values. These properties include thermodynamic state and transport properties needed for structural materials.

In a few cases, stand-alone codes that have been wholly integrated into MELCOR still use properties defined within those codes; a notable example is CORCON, which has been integrated into the Cavity (CAV) package. Also, properties unique to a package, such as those for trace species used in the RadioNuclide (RN) package, are generally modeled within that package. The Core (COR), Fuel Dispersal Interactions (FDI), and Heat Structures (HS) packages principally use the structural materials properties, while the Control Volume Hydrodynamics (CVH), Engineered Safety Features (ESF), Containment Sprays (SPR), and RN packages principally use the fluid transport properties.

The following materials, listed in Table 1.1 with their mnemonic identifiers, are defined in the Material Properties package:

Table 1.1 MP Materials and Mnemonic Identifiers

ZR	Zircaloy	UMETL	Uranium Metal
ZRO2	Zirconium Oxide	GRAPH	Graphite
ZRO2-INT		CON	Concrete
UO2	Uranium Dioxide	ALUM	Aluminum
UO2-INT		AL2O3	Aluminum Oxide
SS	Stainless Steel	CADM	Cadmium
SSOX	Stainless Steel Oxide	SS304	Stainless Steel 304
B4C	Boron Carbide	CS	Carbon Steel
B4C-INT			
AGINC	Silver-Indium-Cadmium		

Remember that mnemonics may be used only as parameters of plot variables and control function arguments, or for abbreviation in the manual. The user must use the full name to define the material (See Table 1.1).

Material SS is a type 347 stainless steel and is typically used in the COR package, whereas material SS304 is a type 304 stainless steel. Materials ZRO2-INT, UO2-INT, and B4C-INT are identical to materials ZRO2, UO2, and B4C, respectively, except for modified melting properties that simulate the reduction in liquefaction temperatures that results from materials interactions.

The following properties listed in Table 1.2 are defined in the MP package:

Table 1.2 Defined Material Properties (Mnemonic)

		Type	Units
1	Enthalpy as a function of temperature (ENH)	Tabular	J/kg
2	Temperature as a function of enthalpy (TMP)	Calculated from ENH	K
3	Specific Heat Capacity as a function of temperature (CPS)	Tabular	J/kg-K
4	Thermal Conductivity as a function of temperature (THC) a. From tables b. From control function c. From tabular function d. From Eucken correlation and Wassijewa equation	 Tabular CF TF Calculated	 W/m-K W/m-K W/m-K W/m-K
5	Density a. Function of temperature (RHO) b. Constant (RHOM)	 Tabular Constant	 kg/m^3 kg/m^3
6	Melting Temperature (TMLT)	Constant	K
7	Latent Heat of Fusion (LHF)	Constant	J/kg

Default values are provided for some, but not all, combinations of materials and physical properties. Table 1.3 summarizes the properties for which default values are available. A *T* indicates that the default function can be changed through user-defined tabular functions (see the Tabular Function (TF) Package Users' Guide) and an MP_PRTF input record. A *C* indicates that the default function can be changed through user-defined constant values input on a MP_PRC record. An *X* indicates that the default function cannot be changed through user input. A blank space indicates that no default is provided, but may be supplied by the user, although in some cases that property for that material may not be used by MELCOR.

Also shown are the mnemonic identifiers used to add new values or alter the default values through user input for those properties which can be changed.

Table 1.3 Default Material Properties, Property Mnemonics, and User Input Capabilities

Property*:	**1**	**3**	**4**	**5a**	**5b**	**6**	**7**
Mnemonic:	ENH	CPS	THC	RHO	RHOM	TMLT	LHF
ZR	T	T	T	T	C	C	C
ZRO2	T	T	T	T	C	C	C
ZRO2-INT	X	X	T	T	C	C	C
UO2	T	T	T	T	C	C	C
UO2-INT	X	X	T	T	C	C	C
SS	T	T	T	T	C	C	C
SSOX	T	T	T	T	C	C	C
B4C	T	T	T	T	C	C	C
B4C-INT	X	X	T	T	C	C	C
AGINC	T	T	T	T	C	C	C
UMETL	T	T	T	T	C	C	C
GRAPH	T	T	T	T	C	C	
CON		T	T	T			
ALUM	T	T	T	T	C	C	C
AL2O3	T	T	T	T	C	C	C
CADM	T	T	T	T	C	C	C
SS304	T	T	T	T	C	C	C
LIAL	T	T	T	T	C	C	C
UAL	T	T	T	T	C	C	C
CS	T	T	T	T	C	C	C

T The default function can be changed using tabular functions and an MP_PRTF input record.
C The default function can be changed using constant values input on an MP_PRC record.
X The default function cannot be changed through user input.
Blank No default provided; user may supply (in some cases MELCOR may not use property for that material).
* See Table 1.2 for a full description of these properties.

1.2 Changes since Version 1.8.6

1.2.1 Adjustment of Melt Point for Interacting Materials

Liquefaction can occur at temperatures significantly below the melt point due to interactions between materials. For example, interaction between ZrO2 and UO2 results in a mixture that is fluid above about 2800 K (compared to the melting temperatures of 3113 K and 2990 K, respectively, for the pure materials). Similarly, although pure B4C melts at 2620 K, interaction with steel produces a mixture that is fluid above about 1700 K. Current practice is to capture these effects by artificially lowering the melt temperatures rather than by enabling the eutectics model in the COR package.

Three new materials, ZRO2-INT, UO2-INT, and B4C-INT, have been added to the material properties library. Their built-in properties match the default properties of the corresponding pure materials, except for having reduced melting temperatures. Unlike other library

materials, the tables relating enthalpy and temperature for these materials are constructed as part of code initialization, based on current values (default or input) for the melting temperature and the latent heat of fusion. Neither these tables, nor the table that defines the specific heat, can be modified directly through user input.

1.3 Changes for F95 Version

The input for all materials that are present in the problem (default, redefined, or new) must be provided in the F95 version format. The user must input the definition record MP_ID for all materials. The other MP records are optional for default materials and materials redefined from the default. However, a complete set of records is required for user-defined properties. The presence of all necessary data for the problem calculation is checked in MELGEN. An error message will be generated if some of the data are absent.

2 User Input

2.1 MELGEN Input

The user may define new materials for the material properties package. This is done by naming the material and defining the properties for that material through a tabular function or a constant value input. If the input material name matches the name of one of the default materials (Table 2.1), the input properties are used instead of the default properties for that material. One set of the following records is required for each new material or for each redefinition of a pre-defined material.

MP_ID – Material Name

Required

This record defines the name of the material. The name may be a default material name or a newly defined name. All input is case-insensitive. Lower-case characters are automatically converted to upper case except for character strings enclosed by single quotes. A dash (-) can be used in place of a space in a default material name.

(1) MATNAM

Default or user-defined material name.

(type = character*24)

Table 2.1 Default Material Names

ZIRCALOY	URANIUM-METAL
ZIRCONIUM-OXIDE	GRAPHITE
ZRO2-INT	CONCRETE
URANIUM-DIOXIDE	ALUMINUM
UO2-INT	ALUMINUM-OXIDE
STAINLESS-STEEL	CADMIUM
STAINLESS-STEEL-OXIDE	STAINLESS-STEEL-304
BORON-CARBIDE	CARBON-STEEL
B4C-INT	
SILVER-INDIUM-CADMIUM	

Example

```
MP_ID SILVER-INDIUM-CADMIUM
```

MP_DELETE – Ignore all records associated with MP_ID

Optional

This record is processed after all records in the input file are read. It will turn off all MP records related to the MATNAM identifier.

(1) MATNAM

Default or user-defined material name.

(type = character*16, default = none, units = none)

Example

```
MP_DELETE  SILVER-INDIUM-CADMIUM
```

MP_PRTF – Property, Tabular Function Pairs

Optional

This record defines the properties by data pairs input as a table. The first field in a pair is a property mnemonic, and the second field is the number of the tabular function to be used for that property. For thermal conductivity only, the second field may also be the name of the control function to be used for that property. In that case, the key control function must be specified on the third optional field; the default is to use tabular function. If a property for a material is given more than once, the last definition is used.

(1) NPRPAR

Number of property data pairs.

(type = integer, default = none, units = none)

The following data are input as a table with length NPRPAR:

(1) NPAR

Data pair index is used for sequencing.

(type = integer, default = none, units = none)

(2) PROP

Property mnemonic. Must be one of the following property mnemonics:

(a) ENH
Enthalpy vs. temperature

(b) CPS
Specific heat vs. temperature

(c) THC
Thermal conductivity vs. temperature

(d) RHO
Density vs. temperature

(type = character*3)

(3) ITBPRP

Name of the tabular function for property, PROP. See the TF Package Users' Guide. Input 'DEFAULT' if the default value is to be used.

(type = character*16, default = 'DEFAULT')

or, <u>for PROP = THC only</u>

Name of the control function for property, PROP. See the CF Package Users' Guide.

(type = character*16, default = none)

(4) CFKEY

Key for control function or tabular function usage. Optional field, used <u>for PROP = THC only</u>.

(a) TF
Use tabular function (default).

(b) CF
Use control function.

(type = character*2, default = TF)

Example

```
MP_ID NEW-CONCRETE             ! redefined
MP_PRTF 4 ! NPAR PROPERTY DEFAULT/TF/CF(may be only for THC)
            1    ENH      DEFAULT
            2    CPS      TF822
            3    THC      TF821   TF
            4    RHO      TF823
```

MP_PRC – Property, Constant Values

Optional

This record defines the constant properties.

(1) RHOM

Constant value for density.

(type = real, defaults = see MP Reference Manual, units = kg/m**3)

(2) TMLT

Constant value for melting temperature.

(type = real, defaults = see MP Reference Manual, units = K)

(3) LHF

Constant value for latent heat of fusion.

(type = real, defaults = see MP Reference Manual, units = J/kg)

(4) WCOMP

Material molecular weight used to calculate properties of molten pool. Only required if used as a COR material. A hyphen "-" or blank in this field will accept the default value.

(type = real, defaults = none, units = kg/mol)

Example

```
MP_ID  URANIUM-DIOXIDE  ! redefined
! density, melting temperature, latent heat of fusion
MP_PRC 10960.00             2800.000             274000.0
```

MP_PVE – Viscosity and thermal expansivity of oxide and metallic molten pools

Optional

This record allows the user to define viscosity and thermal expansivity of oxide and metallic molten pools. Data pairs are input as a table. The first field in a pair is a property mnemonic, and the second field is the value for that property.

(1) NPROPS

Number of property data pairs.

(type = integer, default = none, units = none)

The following data are input as a table with length NPROPS:

(1) NPAR

Data pair index is used for sequencing.

(type = integer, default = none, units = none)

(2) PROP

Property mnemonic.

Must be one of the following property mnemonics:

(a) VIS-OXY-COR-CH
(b) VIS-MET-COR-CH
(c) VIS-OXY-COR-BY
(d) VIS-MET-COR-BY
(e) VIS-OXY-LH-CH
(f) VIS-MET-LH-CH
(g) VIS-OXY-LH-BY
(h) VIS-MET-LH-BY
(i) BET-OXY-COR-CH
(j) BET-MET-COR-CH
(k) BET-OXY-COR-BY
(l) BET-MET-COR-BY
(m) BET-OXY-LH-CH
(n) BET-MET-LH-CH
(o) BET-OXY-LH-BY
(p) BET-MET-LH-BY

(type = character*14, default = none)

optionally followed by

(3) CONST

Constant value for property, PROP. Must be > 0.0. If 0.0 is input, then default model will be used for this property calculation.

(type = real, default = none, units = none)

or

(3) CFNAME

Name of the control function for property, PROP.

(type = character*16, default = none, units = none)

MP_SS – Steel Composition (only for stainless steel [SS])

Optional

This record allows the user to override the default stainless steel composition (74% iron, 18% chromium, 8% nickel, 0% carbon) used by the COR package. If this record is present, all fields must be input. The values will be normalized if required to ensure that the sum of the material fractions equals 1.0.

(1) XFE
Relative mass fraction of iron (must be greater than 0.0).
(type = real, default = 0.74, units = dimensionless)

(2) XCR
Relative mass fraction of chromium.
(type = real, default = 0.18, units = dimensionless)

(3) XNI
Relative mass fraction of nickel.
(type = real, default = 0.08, units = dimensionless)

(4) XCAR
Relative mass fraction of carbon.
(type = real, default = 0.0, units = dimensionless)

Example

```
MP_SS  0.70  0.20  0.10
```

2.2 MELCOR Input

An example was provided following each required MELCOR input record and for many of the optional records.

3 Sensitivity Coefficients

Sensitivity coefficients are not currently used in the MP package.

4 Plot Variables and Control Function Arguments

There are no plot variables or control function arguments currently available in the MP package.

5 Example MELGEN Input

The following are sample MELGEN input records for the MP package. The input for all materials that are present in the problem (default, redefined, or new) must be provided. B4C-INT, ZRO2-INT, and UO2-INT are identical to materials ZRO2, UO2, and B4C, respectively, except for modified melting properties that simulate the reduction in liquefaction temperatures that results from materials interactions. NEW-CONCRETE is a new user-defined material.

```
MP_INPUT
MP_ID ZIRCALOY                 ! default
MP_ID ZIRCONIUM-OXIDE          ! default
MP_ID URANIUM-DIOXIDE          ! default
MP_ID STAINLESS-STEEL          ! default
MP_ID STAINLESS-STEEL-OXIDE    ! default
MP_ID BORON-CARBIDE            ! default
MP_ID SILVER-INDIUM-CADMIUM    ! default
MP_ID URANIUM-METAL            ! default
MP_ID GRAPHITE                 ! default
MP_ID CONCRETE                 ! default
MP_ID ALUMINUM                 ! default
MP_ID ALUMINUM-OXIDE           ! default
MP_ID CADMIUM                  ! default
MP_ID STAINLESS-STEEL-304      ! default
MP_ID CARBON-STEEL             ! default
MP_ID B4C-INT                  ! default
! new or redefined materials:
MP_ID ZRO2-INT                 ! redefined
! density, melting temp., latent heat of fusion for ZRO2-INT
MP_PRC 5600.0  2502.0  707000.0
MP_ID UO2-INT                  ! redefined
! density, melting temp., latent heat of fusion for UO2-INT
MP_PRC 10960.0 2502.0  274000.0
MP_ID NEW-CONCRETE             ! new
MP_PRTF 3
! NPAR PROPERTY DEFAULT/TF/CF(may be only for THC)
  2    CPS      TF822      ! SPECIFIC HEAT VS. TEMP
  3    THC      TF821   TF ! THERMAL CONDUCTIVITY VS. TEMP
  4    RHO      TF823      ! DENSITY VS. TEMP
```

6 Example MELGEN Output (Partial Listing)

```
TOTAL NUMBER OF MATERIALS USED IN PROBLEM =  21

USING TABLES FOR MATERIAL PROPERTIES

Material                 Enthalpy       Specific       Thermal        Viscosity      Density
                                        heat           conductivity
ZIRCALOY                 default        default        default        default        default
ZIRCONIUM-OXIDE          default        default        default        default        default
URANIUM-DIOXIDE          TF72           default        default        default        default
STAINLESS-STEEL          default        default        default        default        default
STAINLESS-STEEL-OXIDE    default        default        default        default        default
BORON-CARBIDE            default        default        default        default        default
SILVER-INDIUM-CADMIUM    default        default        default        default        default
URANIUM-METAL            default        default        default        default        default
GRAPHITE                 default        default        default        default        default
```

```
CONCRETE                 default        TF301          TF300          default        TF302
ALUMINUM                 default        default        default        default        default
ALUMINUM-OXIDE           default        default        default        default        default
CADMIUM                  default        default        default        default        default
STAINLESS-STEEL-304      default        default        default        default        default
CARBON-STEEL             default        TF101          TF100          default        TF102
B4C-INT                  default        default        default        default        default
ZRO2-INT                 default        default        default        default        default
UO2-INT                  default        default        default        default        default
INSULATION-MINFIB        default        TF62           TF63           default        TF61
ZIRCONIUM-DIOXIDE        TF82           default        default        default        default
AIR-GAP-1                default        TF201          TF200          default        TF202

 The following constant properties are redefined.
 If negative, then that property for that material is undefined.

    DENSITY   MELT TEMP.  LAT.HT.FUS.
    KG/M**3       K          J/KG    MATERIAL

  1.0960E+04  2.8000E+03  2.7400E+05 URANIUM-DIOXIDE
  7.7529E+03  1.8109E+03  2.7196E+05 CARBON-STEEL
 -1.0000E+03  0.0000E+00 -1.0000E+03 INSULATION-MINFIB
 -1.0000E+03  0.0000E+00 -1.0000E+03 ZIRCONIUM-DIOXIDE
 -1.0000E+03  0.0000E+00 -1.0000E+03 AIR-GAP-1
```

7 Diagnostic and Error Messages

The error messages in MELGEN pertain to the input and are self-explanatory. An error message is printed when a property uses TF or CF, but such functions are not present in the corresponding package.

Noncondensible Gas (NCG) Package
Users' Guide

Noncondensible gases in the Control Volume Hydrodynamics (CVH) package are modeled as ideal gases. The constant volume heat capacity is approximated as an analytic function of temperature. This document describes user input requirements for specifying a noncondensible gas to be present in a calculation and defining or changing its equation of state.

More detailed information on the constitutive relations and the default values for the existing library of predefined noncondensible gases can be found in the Noncondensible Gas (NCG) Reference Manual.

Contents

1 INTRODUCTION 3

2 MELGEN USER INPUT 3

NCG_ID – Identifier Field 3
NCG_PRP – Material Properties Data 4

3 MELCOR USER INPUT 6

4 SENSITIVITY COEFFICIENTS 7

4.1 NCG Sensitivity Coefficients 7

2090 – Natural Temperature 7
NCG_SC – NCG Sensitivity Coefficient Input Record 7

5 PLOT VARIABLES AND CONTROL FUNCTION ARGUMENTS 8

6 EXAMPLE INPUT 8

7 DIAGNOSTIC AND ERROR MESSAGES 8

1 Introduction

Noncondensible gases in the Control Volume Hydrodynamics (CVH) package are modeled as ideal gases and are characterized by their molecular weight, energy of formation, and specific heat capacity at constant volume, c_v, calculated from the general analytic form

$$c_v(T) = c_{v0} + c_{v1}T + c_{v2}T^2 + c_{v3}T^3 + \frac{c_{vsqrt}}{\sqrt{T}} + \frac{c_{vm1}}{T} + \frac{c_{vm2}}{T^2} \tag{1.1}$$

for the temperature range $T_{low} \le T \le T_{up}$, and held fixed at the boundary value for temperatures below T_{low} or above T_{up}. Each of the coefficients in Equation 1.1, as well as T_{low} and T_{up}, can be specified via user input.

All MELCOR calculations are assumed to involve water in the pool, liquid water in the atmosphere, and water vapor in the atmosphere. Because of this, water in the pool (POOL), which may be two-phase; liquid water in the atmosphere (FOG); and water vapor in the atmosphere (H2O-VAP), are automatically defined in every problem. Noncondensible gases must be defined by the user if desired for a calculation. A library of data for gases of interest is available. The available gases and their associated coefficients for Equation 1.1 are defined in the Noncondensible Gas (NCG) Reference Manual and include hydrogen (H2), deuterium (D2), helium (HE), nitrogen (N2), oxygen (O2), argon (AR), methane (CH4), carbon monoxide (CO), carbon dioxide (CO2), acetylene (C2H2), ethylene (C2H4), ammonia (NH3), nitrogen monoxide (N2O), nitrous oxide (N2O), and user defined gases (GASA through GASJ).

Only the thermal equation of state of noncondensible gases is defined by the NCG package. The thermal equation of state for water is defined by the H2O package and is based on the analytic expression for the Helmholtz function, $\psi(\rho,T)$, that was used to generate the Keenan and Keyes Steam Tables.

2 MELGEN User Input

The user must specify which noncondensible gases are in the calculation and any desired changes in the default coefficients and other constants defining the equation of state (called "material properties"). Liquid and vapor water are always used and do not need to be defined for a calculation. To include a material in the calculation, a record of the following form must be within the MELGEN input.

NCG_ID – Identifier Field

Required

The MELCOR material name is the minimum amount of information required. Library data are used unless the user redefines the material property data on record NCG_PRP. Three materials – water in the pool (may be two-phase), liquid water in the atmosphere (fog), and water vapor in the atmosphere are always used and automatically defined in every problem. All data for user-defined gases must be input.

(1) MNAME

MELCOR material name, e.g., O2 for oxygen.

(type = character*8, default = none, units = dimensionless)

For predefined noncondensible gases, MNAME must be one of the associated coefficients, e.g., O2 for oxygen, H2 for hydrogen, CH4 for methane, etc (see Section 1 for more details on associated coefficients.) Associated coefficients are case sensitive; therefore, Ch4, cH4, or ch4, for example, are not allowed. For user-defined gases, short names are preferable (e.g., UG), but long names with spaces can be used as long as the name is inside quotes and are eight characters at most (e.g., "User Gas" or 'User Gas').

NCG_PRP – Material Properties Data

Optional

This record allows the user to define/redefine material properties data. The material property values are input as a table, each line of which consists of three fields.

(1) NPRP

Number of defined/redefined property values.

(type = integer, default = none, units = dimensionless)

The following data are input as a table with length NPRP:

(1) N

Data string index.

(type = integer, default = none)

(2) MPNAME

Material property name. Refer to the following table of allowed material properties.

(type = character*16, default = none, units = dimensionless)

(3) MPVALUE

Material property value. Refer to the following table for units.

(type = real, default = none, units = depends on property)

MPNAME/VALUE pairs:

Material Property Name	Units	Physical Interpretation
CV0	J/(kg K)	c_{v0} coefficient in $c_v(T)$
CV1	J/(kg K^2)	c_{v1} coefficient in $c_v(T)$
CV2	J/(kg K^3)	c_{v2} coefficient in $c_v(T)$
CV3	J/(kg K^4)	c_{v3} coefficient in $c_v(T)$
CVSQRT	J/(kg $\sqrt{K}$)	c_{vsqrt} coefficient in $c_v(T)$
CVM1	J/kg	c_{vm1} coefficient in $c_v(T)$
CVM2	(J K)/kg	c_{vm2} coefficient in $c_v(T)$
TLOW	K	T_{low} lowest temperature for fit to $c_v(T)$
TUP	K	T_{up} highest temperature for fit to $c_v(T)$
WM	kg/mol	molecular weight
EF	J/kg	e_f energy of formation
SZ	J/(kg K)	s_0 entropy at natural temperature (not used currently)
SIG	Å(10^{-10}m)	Lennard-Jones potential parameter,
EPS	K	Lennard-Jones potential parameter, ε/k

Example

```
! material number is commented and do not need to be input
NCG_ID  N2  ! 4
NCG_ID  CH4 ! 5
NCG_ID  H2  ! 6
! material with redefined data
NCG_ID  O2  ! 7
NCG_PRP 3   ! N MPNAME  MPVALUE
              1 CV0     3000.0
              2 TLOW    200.0
              3 SIG     3.467  ! default value used as example
NCG_ID  CO2 ! 8
NCG_ID  CO  ! 9
```

An additional input switch is used to define the viscosity and thermal conductivity of user-defined gases or to redefine default values of the gases from the NCG default library.

(2) MPNAME

Material property name, THC or VIS (refer to the following table of their physical interpretation).

(type = character*16, default = none, units = dimensionless)

Permitted MPNAME mnemonics:

Material Property Name	Units	Physical Interpretation
THC	K	thermal conductivity
VIS	kg/mol	viscosity

(3) MPSWITCH

Additional material property input switch.

(a) C-E

To specify the use of Chapman-Enskog relationships for THC or VIS properties. If Chapman-Enskog relationships are specified, the Lennard-Jones potential parameters SIG and EPS may be defined by the user or changed from the defaults.

(b) CF

To use the control function for property (required field CFNAME below).

(c) TF

To use the tabular function for property (required field TFNAME below).

(type = character*3, default = none, units = dimensionless)

When the user specifies MPSWITCH to be either CF or TF, the control or tabular function name for the property must be included.

(4) CFNAME

Name of the control function for property.

(type = character*16, default = none, units = none)

or

(4) TFNAME

Name of the tabular function for property.

(type = character*16, default = none, units = none)

Example

```
NCG_ID  UserGas
NCG_PRP 2   ! N MPNAME
              1 THC     CF       THC_UserGas
              2 VIS     C-E
```

3 MELCOR User Input

Currently, the NCG database and models cannot be changed via MELCOR input.

4 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. Their use is described in Section 6 of the MELCOR Executive (EXEC) Users' Guide.

4.1 NCG Sensitivity Coefficients

Only one sensitivity coefficient is used. It is the temperature T_n that appears in the integral of specific heat $c_v(T)$ that is used to determine the specific internal energy and enthalpy (see the NCG Reference Manual).

2090 – Natural Temperature

(1) The lower temperature limit used in the integral of $c_v(T)$ to calculate specific internal energy and enthalpy.

(default = 298.15, units = K, equiv = none)

NCG_SC – NCG Sensitivity Coefficient Input Record

Optional

This record allows the user to change the default NCG sensitivity coefficient value to a user-defined value. This record is not required.

(1) NUMSTR

Number of data strings.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the NCG sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only–no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

5 Plot Variables and Control Function Arguments

There are no plot variables or control function arguments for the NCG package.

6 Example Input

Water in the pool (may be two-phase), liquid water in the atmosphere (fog), and water vapor in the atmosphere are always used in the problem. The example input shown below defines two additional materials, hydrogen with default properties and oxygen with a modified value for c_{v0} of 3000.

```
NCG_ID  O2
NCG_PRP 1  ! N MPNAME  MPVALUE
             1 CV0     3000.0
NCG_ID  H2
NCG000 O2     5      CV0    3000.
NCG010 H2     4
```

7 Diagnostic and Error Messages

Diagnostic and error messages can originate from incorrect user input. The messages will be self-explanatory.

Passive Autocatalytic Recombiner (PAR) Package Users' Guide

The MELCOR Engineered Safety Features (ESF) package models the phenomena for the various engineered safety features in a nuclear power plant. The Passive Autocatalytic Recombiner (PAR) package is a subpackage of the ESF package and calculates the removal of hydrogen from the atmosphere due to the operation of passive hydrogen reaction devices.

This users' guide provides basic information needed to run the PAR model with other MELCOR packages, including a detailed explanation of the user input and package output for MELGEN and MELCOR. Required and optional input, control function arguments, plot variables, and error messages are all included.

More detailed information on the phenomenological modeling and numerical solution schemes implemented in the PAR package can be found in the PAR Package Reference Manual.

Contents

1 INTRODUCTION 3

2 INPUT REQUIREMENTS 3

2.1 MELGEN Input 3

PAR_ID – Hydrogen Recombiner Name 3
PAR_ICI – Hydrogen Recombiner Interface and Control 4
PAR_PRM – Hydrogen Recombiner Parameters 5
PAR_CLD – Hydrogen Recombiner Combustion Limit Data 6

2.2 MELCOR Input 6

3 SENSITIVITY COEFFICIENTS 7

4 PLOT VARIABLES 7

5 CONTROL FUNCTION ARGUMENTS 7

6 EXAMPLE INPUT 8

7 PAR MODEL OUTPUT 9

8 DIAGNOSTIC AND ERROR MESSAGES 9

APPENDIX 1 10

1 Introduction

The MELCOR Engineered Safety Features (ESF) package models the thermal-hydraulic behavior of various engineered safety features in nuclear power plants. One such device is the passive autocatalytic hydrogen recombiner (PAR).

The MELCOR PAR model is based on the Fischer model (see the PAR Package Reference Manual), which is a parametric model developed for the most common PAR design. The user input provides correlation coefficients for the general mathematical form of the model. These coefficients are used by the code to calculate the total gas flow rate through a PAR unit. From the PAR gas flow rate together with user-provided PAR efficiencies, transient relaxation times, delay times, and the internally calculated hydrogen mole fractions, a per-PAR-unit hydrogen reaction rate is calculated. This rate is then multiplied by the current timestep and the user-provided number of active PAR units to determine the change in hydrogen, oxygen, and steam masses. These differential masses are then passed to the Control Volume Hydrodynamics (CVH) package as source/sink terms.

2 Input Requirements

This section includes the input requirements for the MELCOR PAR package, including a short description of the input quantities and their units and default values, if any. Further description of the input variables and their meaning within the models can be found in the PAR Package Reference Manual.

Input for the PAR model must be within the data block for the ESF package that begins with the character string ESF_INPUT. Multiple hydrogen recombiner types can be specified, and input is grouped into sets for each hydrogen recombiner modeled, identified by record PAR_ID.

2.1 MELGEN Input

PAR_ID – Hydrogen Recombiner Name

Required

This record specifies a user-supplied name for the hydrogen recombiner. This record is required. The following character field (limited to 16 characters) must be present.

(1) FPRNAM

PAR name.

(type = character*16, default = none)

Example

```
PAR_ID  NISPAR1
```

PAR_ICI – Hydrogen Recombiner Interface and Control

Required

This record specifies the control volume to which the PAR unit is interfaced, a control function/flag to switch between the basic model and a user-provided model, and a control function/flag to switch between the constant efficiency and a user-provided variable efficiency. This record is required, but only the first field must be present. The second and third fields are input pairwise as FLAG – CF NAME (if needed).

(1) IPAR

Hydrogen recombiner control volume name.

(type = character*16, default = none, units = none)

(2) IPROPT

Flag for selection of the hydrogen recombiner flow model. If this number is specified as zero or FISCHER, the basic Fischer model will be used. Otherwise, the identifier name of a control function that provides the PAR unit total gas volumetric flow rate must be input.

(a) 0 or FISCHER
The basic Fischer model will be used.

(b) -1 or CF
Control function provides the PAR unit total gas volumetric flow rate.

and if IPROPT = CF or -1, then

(2a) CFNFLO

The identifier name of a control function that provides the PAR unit total gas volumetric flow rate.

(type = character*16, default = none)

(3) IETAPR

Flag for selection of the hydrogen recombiner efficiency model. If this number is specified as zero or EPAR, a constant efficiency (EPAR, provided on the PAR_PRM record) will be used for the PAR efficiency. Otherwise the identifier name of a control function that provides the PAR efficiency must be input.

(type = integer / character*4, default = 0, units = none)

(a) 0 or EPAR
Constant efficiency will be used.

(b) -1 or CF
Control function provides the PAR efficiency.

and if IETAPR = CF or -1, then

(3a) CFNEFF

The identifier name of a control function that provides the PAR efficiency.

(type = character*16, default = none)

Example

```
PAR_ICI  CVol100        CF  CF101        CF  CF102
```

PAR_PRM – Hydrogen Recombiner Parameters

Optional

This record specifies the Fischer model flow rate correlation coefficients, the transient effect parameters, and the PAR multiplicity. This record is optional, but if any of the fields require changes from default values, the entire set of six parameters must be supplied.

(1) APAR

Hydrogen recombiner correlation coefficient.

(type = real, default = 0.67, units = m^3/s)

(2) BPAR

Hydrogen recombiner exponential parameter.

(type = real, default = 0.307, units = none)

(3) EPAR

Hydrogen recombiner efficiency.

(type = real, default = 0.85, units = none)

(4) TAUPAR

Hydrogen recombiner transient relaxation time.

(type = real, default = 1800.0, units = s)

(5) TPARD

Hydrogen recombiner operation delay time.

(type = real, default = 0.0, units = s)

(6) FPARD

Number of hydrogen recombiners of this type. Note that this does not have to be a whole number of units. The degraded operation of one or more units can be simulated by using a fraction of a PAR unit.

(type = real, default = 1.0, units = none)

Example

```
PAR_PRM   0.75   0.300   0.75   1800.0   0.0   20.0
```

PAR_CLD – Hydrogen Recombiner Combustion Limit Data

Optional

This record specifies the on/off reactant concentration limits of the PAR unit. The first two fields specify the minimum hydrogen mole fraction for which the unit will start operating (startup point) and the minimum hydrogen mole fraction to which the unit will reduce the hydrogen concentration (shutoff point). The third and fourth fields specify the same values for the oxygen concentration limits. This record is optional, but if any of the fields require changes from default values, the entire set of four parameters must be supplied.

Note: Care must be exercised to ensure that the shutoff concentrations are always less than the startup concentrations. Also, due to lack of data, the default values used here are not technically defensible.

(1) HPAR0
Minimum H_2 mole fraction for PAR startup
(type = real, default = 0.02, units = none)

(2) HPARR
Minimum H_2 mole fraction for PAR shutdown
(type = real, default = 0.005, units = none)

(3) OPAR0
Minimum O_2 mole fraction for PAR startup
(type = real, default = 0.03, units = none)

(4) OPARR
Minimum O_2 mole fraction for PAR shutdown
(type = real, default = 0.005, units = none)

Example

```
PAR_CLD   0.01   0.001   0.02   0.001
```

2.2 MELCOR Input

No input for the H_2 recombiner model is processed during MELCOR execution.

3 Sensitivity Coefficients

Because the required model parameters are all accessible through the input record, sensitivity coefficients for the hydrogen recombiner model are not required.

4 Plot Variables

The plot variables currently included in the PAR model are listed below, along with a brief description. Note that control function arguments (some are identical in definition to these plot variables but different in format) are described in the following section.

ESF-PAR-DMH2.n	Per PAR unit H_2 removal rate for PAR index n. (units = kg/s)
ESF-PAR-IMH2.n	Total H_2 removed for all FPARD units of PAR index n. (units = kg)
ESF-PAR-DVOL.n	Per PAR unit total gas flow rate for PAR index n. (units = m^3/s)
ESF-PAR-IVOL.n	Total volume of gas processed in all FPARD units of PAR index n. (units = m^3)
ESF-PAR-TOUT.n	Outlet gas temperature for PAR index n. (units = K)
ESF-PAR-FMOL.n	Outlet gas H_2 mole fraction for PAR index n. (units = dimensionless)

5 Control Function Arguments

The control function arguments currently included in the PAR model are listed below, along with a brief description. Note that plot variables (some are identical in definition to these control function arguments but different in format) are described in the previous section.

ESF-PAR-DMH2(NamePAR)	Per PAR unit H_2 removal rate for PAR name NamePAR. (units = kg/s)
ESF-PAR-IMH2(NamePAR)	Total H_2 removed for all FPARD units of PAR name NamePAR. (units = kg)

ESF-PAR-DVOL(NamePAR)	Per PAR unit total gas flow rate for PAR name NamePAR. (units = m^3/s)
ESF-PAR-IVOL(NamePAR)	Total volume of gas processed in all FPARD units of PAR name NamePAR. (units = m^3)
ESF-PAR-TOUT(NamePAR)	Outlet gas temperature for PAR name NamePAR. (units = m^3/s)
ESF-PAR-FMOL(NamePAR)	Outlet gas H_2 mole fraction for PAR name NamePAR. (units = m^3)

6 Example Input

The following are sample MELGEN input records for the PAR model. No MELCOR input records are necessary to run the PAR model.

```
ESF_INPUT
PAR_ID  'NISPAR1'
!          IPAR    IPROPT          IETAPR
PAR_ICI   CV100   CF 'PAR-FLOW'   CF  'PAR-EFF'
!         APAR   BPAR    EPAR   TAUPAR  TPARD FPARP
PAR_PRM   0.75   0.300   0.75   1800.0   0.0   20.0
!         HPAR0  HPARR   OPAR0  OPARR
PAR_CLD   0.01   0.001   0.02   0.001
...
CF_INPUT
!
! CONTROL FUNCTION FOR H2 RECOMBINER FLOW RATE
!
! PAR GAS FLOW RATE USING FISHER MODEL (NO TRANSIENT)
! FOR CONTROL VOLUME CV100 & HYDROGEN.
!
CF_ID 'PAR-FLOW' POWER-R
CF_SAI 0.67 0.0
CF_MSC 0.307
CF_ARG    1 ! NARG CHARG             ARSCAL  ARADCN
                1 CVH-X(CV100,H2)     1.0      0.0
!
! CONTROL FUNCTION FOR H2 RECOMBINER EFFICIENCY
!
! PAR EFFICIENCY USING A CONSTANT
!
CF_ID 'PAR-EFF' EQUALS
CF_SAI  1.0   0.0
CF_ARG    1 ! NARG CHARG      ARSCAL   ARADCN
                1 EXEC-TIME    0.0      0.75
```

7 PAR Model Output

In general, the PAR model output is self-explanatory. The hydrogen removal rate, gas flow rate, total reacted hydrogen, total volume of processed gas, PAR outlet gas temperature, and H_2 mole fraction are output for each PAR unit.

8 Diagnostic and Error Messages

Diagnostics and error messages generated during MELGEN are concerned with input processing and are generally self-explanatory. Currently, no messages are generated during MELCOR execution.

Appendix 1

Melcor 2.1	PTFRead	Comments
ESF-PAR-DMH2(NamePAR) NamePAR – the PAR name.	ESF-PAR-DMH2.n n – the PAR index.	Per PAR unit H_2 removal rate for PAR name NamePAR (index n).
ESF-PAR-IMH2(NamePAR) NamePAR – the PAR name.	ESF-PAR-IMH2.n n – the PAR index.	Total H_2 removed for all FPARD units of PAR name NamePAR (index n).
ESF-PAR-DVOL(NamePAR) NamePAR – the PAR name.	ESF-PAR-DVOL.n n – the PAR index.	Per PAR unit total gas flow rate for PAR name NamePAR (index n).
ESF-PAR-IVOL(NamePAR) NamePAR – the PAR name.	ESF-PAR-IVOL.n n – the PAR index.	Total volume of gas processed in all FPARD units of PAR name NamePAR (index n).
ESF-PAR-TOUT(NamePAR) NamePAR – the PAR name.	ESF-PAR-TOUT.n n – the PAR index.	Outlet gas temperature for PAR name NamePAR (index n).
ESF-PAR-FMOL(NamePAR) NamePAR – the PAR name.	ESF-PAR-FMOL.n n – the PAR index.	Outlet gas H_2 mole fraction for PAR name NamePAR (index n).

RadioNuclide (RN) Package
Users' Guide

This document includes a brief description of the models employed in the RadioNuclide (RN) package, detailed descriptions of the input format, discussion of the output, sensitivity coefficients, and plot variables, and example input for a typical plant calculation. Details on the various models employed in the RN package can be found in the RN Package Reference Manual.

Contents

CONTENTS **2**

LIST OF TABLES **6**

1 INTRODUCTION **7**

1.1 Changes since Version 2.0 7

1.2 Changes since Version 1.8.6 8

1.2.1 Normalization of RN inventories 8

1.2.2 Extension of the CORSOR-Booth Release Model to a Second Fuel Type 9

1.2.3 Additional Release Model Options 9

1.2.4 More Flexible Aerosol Filter Model 9

1.2.5 Treatment of MAEROS Aerosol Coefficients 9

1.2.6 Plot Variables and Control Function Arguments 10

2 DESCRIPTION OF MODELS **10**

2.1 General Framework 10

2.2 Initial Radionuclide Inventories 12

2.3 Release of Radionuclides 12

2.4 Aerosol Dynamics 13

2.5 Condensation/Evaporation and Hygroscopic Behavior 14

2.6 Decay Heat Distribution 15

2.7 ESF Models 15

2.8 Fission Product Chemistry 16

2.9 Iodine Pool Chemistry Model 16

3 USER INPUT **17**

3.1 MELGEN Input 17

3.1.1 General Control, Options, and Mappings 17

RN1_INPUT – Start record for RN1 package user-input 17
RN1_DFT – RN Package Default Scheme 18
RN1_DIM – Dimension 19
RN1_HM – Activate Hygroscopic Model 20
RN1_DCHNORM - Class Normalization 20
RN1_CRCL – Core Material to RN Class Map 22
RN1_CLVN – RN Class to VANESA Group Map 23
RN1_VNCL – VANESA Group to RN Class Map 24

RN1_DIFMAP - Maps tracked HTGR species to the RN classes 26

3.1.2 Initial Radionuclide Inventories 27

RN1_FPN – Initial Core Fuel and Cavity Radionuclide Inventories 27
RN1_GAP – Initial Fuel-Cladding Gap Inventory Fractions 30
RN1_AG – Initial Aerosol Masses in Atmosphere 32
RN1_ADFG – Aerosol Deposition Model Deactivation Flag 33
RN1_AL – Initial Aerosol Masses in Pool 33
RN1_VG – Initial Fission Product Vapor Masses in Atmosphere 34
RN1_VL – Initial Fission Product Vapor Masses in Pool 35
RN1_DSCAL - Transport Deposition Scaling 36

3.1.3 Release Model Parameters 37

RN1_FP00 – Core Release Model 37
RN1_BOOTH00 – Identification of Cs class(es) for the CORSOR-Booth release model 38
RN1_BOOTH01 – Identification of classes associated with alternate fuel 39
RN1_GAP00 – Gap Release Temperature 41
RN1_CLS – Parameters for Class Combination at Release 42

3.1.4 Aerosol Modeling Parameters 47

RN1_ASP – Aerosol Sectional Parameters 47
RN1_ACOEF – Aerosol Coefficients 48
RN1_PT – Conditions for Aerosol Coefficients 49
RN1_MS00 – Miscellaneous Aerosol Dynamics Constants 50
RN1_MS01 – Miscellaneous Aerosol Dynamics Constants 50
RN1_CC – Class/Component Map 51
RN1_DS – Radionuclide Deposition Surfaces 52
RN1_SET – Flow-through Areas for Intervolume Transport 53
RN1_AS – Aerosol Source 43
RN1_AS01 – Sectional Distribution Parameter 44
RN1_AR – Aerosol Resuspension Parameters 57

3.1.5 Condensation/Evaporation Models 58

RN1_ACD – Aerosol Condensation Index 58
RN1_VS – Vapor Source 46

3.1.6 Decay Heat Distribution 59

RN1_DHL – Length for Absorption of Decay Betas 59
RN1_DHV – Decay Heat Split for Control Volumes 60
RN1_DHVS – Control Volume Decay Heat Split to Other Components 61
RN1_DHS – Decay Heat Split for Heat Structures 63
RN1_DHSS – Surface Decay Heat Split to Other Components 64

3.1.7 ESF Parameters 66

RN2_INPUT – Start record for RN2 Package input 66
RN2_PLS – Pool Scrubbing 66
RN2_FLT – Filter Parameters 67

RN2_FCL – Filter Class Parameters 69
RN2_FDA – Decontamination Factor as a Function of Aerosol Size 70
RN2_FDC - Decontamination Factor as a Function of RN Class and Aerosol Size 72
RN2_FIM – Initialization of Filter RN Mass 73
RN2_FAC – Performance Characteristics of Aerosol Filters 74
RN2_FVC – Performance Characteristics of Charcoal (Vapor) Filters 75
RN2_FTD – Thermal Desorption Data for Charcoal Filters 75
RN2_FRD – Radiolytic Desorption Data for Charcoal Filters 77
RN2_FCD – Charcoal Combustion Data for Charcoal Filters 78
RN2_CNV – Options 79

3.1.8 Radionuclide Chemistry 79

RN1_RCT – Class Reaction Information 79
RN1_RCT00 – *From* Class Specification 80
RN1_RCT01 – *To* Class Specification 80
RN1_RCT02 – Translation Array 81
RN1_RCT03 – Control Volumes in which the reaction may occur 82
RN1_RCT04 – States where the reaction can occur 82
RN1_RCT10 – Forward Reaction Mass Transfer Data 84
RN1_RCT11 – Backward Reaction Mass Transfer Data 85
RN1_RCT20 – Forward Reaction Energy Transfer Data 85
RN1_RCT21 – Backward Reaction Energy Transfer Data 86
RN1_TRN – Class Transfer Information 87
RN1_TRN00 – *From* Class Specification 87
RN1_TRN01 – *To* Class Specification 88
RN1_TRN02 – Translation Array 88
RN1_TRN03 – Control Volumes in which the transfers may occur 89
RN1_TRN04 – States where the transfers can occur 90
RN1_TRN10 – Mass Transfer Data 92
RN1_TRN11 – Energy Transfer Data 93

3.1.9 Chemisorption 94

RN1_CAF – Chemisorption Activation Flag 94
RN1_CAON – Chemisorption Class Activation Flags 94
RN1_CACM – Chemisorption Class to Radionuclide Vapor Mapping 95

3.1.10 Iodine Pool Model 96

RN1_IOD01 – Iodine model activation flags 96
RN1_IOD02 – Iodine pool model start time 97
RN1_IODCV – Iodine pool model Control Volume 98
RN1_PH – pH Calculation Types 98
RN1_DOS – Dose input 99
RN1_CB – Cable mass input 101
RN1_WSC – Surface coating on walls 102
RN1_IOP – Iodine pool species to be output on plot file. 103

3.1.11 Flashing-Jet Model 104

RN1_JI – Impacting jet model 104
3.1.12 Plate Film Entrainment Model 106
RN1_ENT – Entrainment from films on HS surfaces 106
3.1.13 Lower Head Molten Pool Model 107
RN1_MPCR – Molten pool concentration ratios of radionuclides 107
3.2 MELCOR Input 108
RN1_EDTFLG – RN Edit Flags 108
RN1_IOD02 – Iodine pool model start time 109
RN1_VISUAL – Special-purpose output for visualization 109

4 SENSITIVITY COEFFICIENTS 110

7000 – Differential Equation Convergence 111
7001 – Aerosol Coefficient Criteria 112
7002 – Fission Product Decay Beta Range 112
7003 – Parameters for Interpolation of Aerosol Coefficients 112
7100 – COR Material Release Multipliers 112
7101 – CORSOR Coefficients 113
7102 – CORSOR-M Coefficients 115
7103 – CORSOR-Booth Class Scaling Factors: Nominal Values 115
7104 – Release Surface-to-Volume Ratio 116
7105 – Modification of Release Rates 116
7106 – CORSOR-Booth Transient Release Parameters for Cesium 117
7107 – CORSOR-Booth Class Scaling Factors: Oxidation Modified 117
7110 – Vapor Pressure 118
7111 – Vapor Diffusivity Constants 121
7120 – Class Molecular Weights 121
7130 – Oxidation-Based Release Coefficients for Metallic Fuels 122
7131 – Birney Non-Oxidation Release Coefficients for Metallic Fuel 123
7132 – Arrhenius Non-Oxidation Release Coefficients for Metallic Fuel 124
7135 – Noble Gas Release on Failure of Metallic Fuel 125
7136 – Solubility of RN Classes in Water Films 126
7140 – Release from Molten U-Al Pools 126
7141 – Solubility of Classes in Al-U Alloy 127
7142 – Debris Particle of Average Surface Area 128
7143 – Molten Fraction Criterion for Release from U-Al Pools 128
7144 – Temperature Criterion for Release from Intact Fuel 128
7150 – SPARC-90 Model Parameters 128
7151 – SPARC-90 Globule Size Correlation 130
7152 – SPARC-90 Bubble Size/Shape Model 130
7153 – SPARC-90 Bubble Rise Velocity Model 131
7154 – SPARC-90 Swarm Velocity Model 131
7155 – SPARC-90 Particle Impaction Model 132
7156 – SPARC-90 Solute Ionization Correlations 133
7157 – SPARC-90 Settling Velocity Correlation 134

7158 – SPARC-90 HOI Correlation 135
7159 – SPARC-90 I2 Chemistry Model Parameters 136
7160 – Chemisorption 137
7170 – Hygroscopic Aerosol Parameters 138
7180 – Iodine Pool Model Mass Transfer Parameters 140
7181 – Iodine Pool Chemistry Iteration Parameters 141
7182 – Iodine Pool Chemistry Activation Limits 141
RN1_SC – RN1 Sensitivity Coefficients Input 142
RN1_CSC – RN Class Sensitivity Coefficients Input 143

5 PLOT VARIABLES 144

6 CONTROL FUNCTION VARIABLES 152

7 EXAMPLE INPUT 163

8 ERROR MESSAGES 166

APPENDIX 1 168

List of Tables

Table 1.1 Required and Recommended Classes for Different Models 8
Table 2.1 RN Class Compositions 11
Table 3.1 Aqueous Species Names 103
Table 4.1 Default Hygroscopic Properties of RN Class Compounds. 140

1 Introduction

The RadioNuclide (RN) package models the behavior of fission product aerosols and vapors and other trace species, including release from fuel and debris, aerosol dynamics with vapor condensation and revaporization, deposition on structure surfaces, transport through flow paths, and removal by engineered safety features. The package also allows for simplified chemistry controlled by the user.

Boundary conditions for the various models are obtained from other MELCOR packages: fluid conditions are obtained from the Control Volume Hydrodynamics (CVH) package, fuel and debris temperatures are obtained from the Core (COR) and Cavity (CAV) packages, and structure surface temperatures are obtained from the Heat Structures (HS) package. The COR and CAV packages also provide information regarding bulk debris relocation, allowing the RN package to perform relocation of unreleased fission products. Likewise, advection of radionuclides between control volumes is done using CVH flows, and wash-off of radionuclides deposited on heat structures is determined from drainage of water films calculated by the HS package. The RN package determines decay heat power for current radionuclide inventories from the Decay Heat (DCH) package when requested by both of these packages.

This document includes a brief description of the models employed in the RN package, detailed descriptions of the input format, discussion of the output, specification of the sensitivity coefficients, and plot variables, and example input for a typical plant calculation. Details on the various models employed in the RN package can be found in the RN Package Reference Manual.

1.1 Changes since Version 2.0

In MELCOR 2.0, 17 fixed default classes (Xe, Cs, Ba, etc.) are automatically defined in the code. In addition to the default classes user-defined classes can be used. In MELCOR 2.0, an arbitrary number of user-defined classes can be used. When the RN package is active, the least number of classes used in MELCOR 2.0 is equal to 1 (H2O). The classes with their names must be defined in DCH package input. An arbitrary number of additional classes can be added by the user.

The number of required classes depends on the specific models which can be activated in the RN1 package, namely the iodine pool model, the boron chemistry model, and the silver release model as shown in Table 1.1.

For the iodine pool model, six additional deposition classes are defined automatically inside the code if the model is activated. These classes are used for deposited iodine and methyl iodide on steel and painted surfaces, and for hydrochloric and nitric acid.

The classes used are

(1) Deposited methyl iodine

(2) Chemically bound methyl iodine

(3) Deposited iodine

(4) Chemically bound iodine

(5) Nitric acid

(6) Hydrochloric acid

Table 1.1 Required and Recommended Classes for Different Models

Model	Iodine pool model	Boron chemistry model	Silver release model
Required classes	H2O, CH3I, IM, HCL, HNO3, I2, CSI, CS, FE, AG	H2O, H3B3O6, BH3, B2H6, BOH, B(S), C(S), HBO2, BO2	H2O, AG-CR, IN-CR, CD-CR
Recommended optional classes	HBO2, LIBO, HPO4, AGI	—	—

1.2 Changes since Version 1.8.6

A number of extensions and enhancements to the models were made to the RN package in MELCOR 1.8.6 and apply to MELCOR 2.x. They are summarized in the following subsections.

1.2.1 Normalization of RN inventories

The initial mass of class URANIUM (class number 10 by default, UO2) in RN1 is derived from the mass of UO2 in the COR package, rather than from RN1 input. If the RN1_FPN input tabular records allocate 100% of the reference core in the DCH package to the actual core, all class masses and decay heats should match, except for class URANIUM. Because normalization of total decay power is applied to the reference core, it cannot compensate for the discrepancy.

The initialization has been changed. By default, the mass (kg) and decay heat (W/kg) of class URANIUM in the DCH reference core are now modified to reconcile the mass with the total initial mass defined by RN1 input, without changing the total decay heat associated with the class. In most cases, this results in matching of all class masses and decay heats between RN1 and DCH.

Input of RN1_DCHNORM with keyword NONE (or, at least, with none of the fields containing UO2) is necessary for calculations where DCH is active but there is no UO_2 mass in the problem or no UO2 class is set. Otherwise, in the first case MELGEN writes the following error message:

```
ERROR IN SUBROUTINE DCH_DCHMCL
ATTEMPT TO NORMALIZE CLASS WITH NAME xxx
TO NON-POSITIVE CLASS MASS 0.0E+00
```

In the second case, MELGEN writes the following error message:

```
Error in table RN1_DCHNORM
THE CLASS NAME UO2 IS NOT FOUND
```

and no restart file is created in both cases.

1.2.2 Extension of the CORSOR-Booth Release Model to a Second Fuel Type

There are cases where a reactor core can contain two different types of fuel rods. While it is probably adequate for most purposes to average their properties, the different structures may result in very different release rates for fission products. As a first step towards allowing this effect to be captured, the CORSOR-Booth release model was extended to recognize two different types of fuel. To accommodate this, sensitivity coefficient array 7106 must now be specified with two indices. By default, release rates for the two fuel types are identical.

The CORSOR-Booth model is based on scaling release rates calculated for Cs for application to all RN classes. Thus, if there are two types of fuel, the Cs classes associated with them must be different. As currently implemented, it is necessary to assign all fission products in the second, alternate fuel type to classes distinct from those for the normal fuel.

1.2.3 Additional Release Model Options

There is a conceptual problem in the CORSOR-Booth release model in previous MELCOR versions. It was implemented based on published code from Battelle Columbus, which attempts to account for the effects of gas-phase diffusion. Part of the calculation involves inferring a correction to the diffusion rate for release of cesium from fuel grains. However, this calculation is based on current thermo-hydraulic conditions, not on those that existed during the experiments used to correlate the fuel-grain diffusion coefficient, and the correction therefore has no connection to reality. In particular, the resulting release rate for cesium is independent of thermo-hydraulic conditions. In rare instances, the correction can lead to totally unrealistic results.

We have, therefore, implemented an alternate version, in which the gas-phase diffusive resistance is added to a fuel grain resistance that is calculated using the correlated diffusion coefficient without correction.

1.2.4 More Flexible Aerosol Filter Model

The aerosol filter model has been extended to allow specification of decontamination factor by particle size as well as by class. Each decontamination factor may now be defined by either a constant or a control function.

1.2.5 Treatment of MAEROS Aerosol Coefficients

The aerosol kernel used in MAEROS is a function of temperature and pressure. The values used are obtained by interpolation of values pre-calculated at pressure and temperature points defined by the RN1_PT input record. MELCOR 1.8.5 used bilinear interpolation, which can lead to significant errors if the range of temperature and/or pressure is great.

We have found that much more accurate values can be obtained using a power law interpolation

$$\text{Coefficient} \sim P^{PEXP}\ T^{TEXP}$$

The default value of PEXP, -1, was inferred from the form of the equations. That for TEXP, 0.5, was based on optimization in a least squares sense for a few specific cases, but the sensitivity is not very great.

The option of reading pre-calculated aerosol coefficients from a file to save execution time is almost never used—the time involved is negligible on modern computers. Thus, the ACOEFF file, written to contain calculated coefficients, is almost never used, and its creation can cause an unexpected request for permission to overwrite an existing file. The default treatment and user input options on the RN1_ACOEF record have been changed to reflect modern practice.

1.2.6 Plot Variables and Control Function Arguments

New variables include total released class masses (plot), total airborne class masses by volume type (control function argument), and radioactive aerosol and vapor class masses on filters (control function argument). Total aerosol and vapor class masses on filters are now available as plot variables and control function arguments. There is a new option to specify releases as radioactive in element form, total in elemental form, or total as a compound. Documentation has been corrected to show that aerosol size parameter values are also available as control function arguments.

Masses of radioactive masses transported through flow paths are available as plot variables and/or control function arguments for paths that have been designated as MACCS2 release paths. This capability is described in the Executive (EXEC) Package Users' Guide.

2 Description of Models

The RN package in MELCOR calculates the release and transport behavior of fission product vapors and aerosols. The models and concepts included in the RN package are discussed in detail in the RN Package Reference Manual. Only a brief overview is included in this section as a guide to understanding user input requirements.

2.1 General Framework

The RN package operates on the basis of material classes, which are groups of elements that have similar chemical properties. The number of classes is specified on the RN1_DIM input record, with a default of 17 classes in MELCOR 2.0 and 16 classes in MELCOR 1.8.6. The grouping of the different elements into these classes is shown in Table 2.1. Classes are generally referred to by their class name or representative element. Combination of masses in these classes upon release to form compounds in other classes, such as Cs + I to CsI or Cs + Mo to CsM, is permitted subject to stoichiometric constraints (e.g., excess Cs is retained in the Cs class). For the RN

package, the classes must be in numerical order without any gaps when using MELCOR 1.8.6. This requirement is removed beginning with MELCOR 2.0 since specific class names are associated with RN in MELCOR 2.0.

Some models in the RN package use groupings of elements different from the groupings defined in Table 2.1. Transfers of masses between various models must, therefore, use mapping strategies, which are described in the RN Package Reference Manual. These mappings may be changed with the input records described in Section 3.1.1.

Warning: If a class is redefined from the default values, or if a new class is added, all of the properties, including mappings, should be evaluated and possibly redefined through the RN sensitivity coefficients. Default values for these properties are defined based on the default elements in each class. Whether default values are appropriate when classes are modified must be determined by the user. Note that the DCH package might also have to be redefined in a consistent manner.

Table 2.1 RN Class Compositions

Class	Class Name	Chemical Group	Representative	Member Elements
1	XE	Noble Gas	Xe	He, Ne, Ar, Kr, Xe, Rn, H, N
2	CS	Alkali Metals	Cs	Li, Na, K, Rb, Cs, Fr, Cu
3	BA	Alkaline Earths	Ba	Be, Mg, Ca, Sr, Ba, Ra, Es, Fm
4	I2	Halogens	I_2	F, Cl, Br, I, At
5	TE	Chalcogens	Te	O, S, Se, Te, Po
6	RU	Platinoids	Ru	Ru, Rh, Pd, Re, Os, Ir, Pt, Au, Ni
7	MO	Early Transition Elements	Mo	V, Cr, Fe, Co, Mn, Nb, Mo, Tc, Ta, W
8	CE	Tetravalent	Ce	Ti, Zr, Hf, Ce, Th, Pa, Np, Pu, C
9	LA	Trivalents	La	Al, Sc, Y, La, Ac, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Am, Cm, Bk, Cf
10	UO2	Uranium	UO_2	U
11	CD	More Volatile Main Group	Cd	Cd, Hg, Zn, As, Sb, Pb, Tl, Bi
12	AG	Less Volatile Main Group	Ag	Ga, Ge, In, Sn, Ag
13	BO2	Boron	BO_2	B, Si, P
14	H2O	Water	H_2O	H_2O

Class	Class Name	Chemical Group	Representative	Member Elements
15	CON	Concrete	CON	- – -
16	CSI	Cesium iodide	CsI	CsI
17	CSM	Cesium Molybdate	CsM[1]	CsM[1]

[1]Cesium Molybdate (Cs2MoO4) is represented in MELCOR as CSM in order to satisfy the three character naming limitation in MELCOR,

2.2 Initial Radionuclide Inventories

Initial inventories and distributions of radionuclides must be specified for the core, for the cavity, and for control volume pools and atmospheres. (Inventories for some locations may be zero initially.) Radionuclide masses can be distributed among core cells according to radial and axial decay heat power profiles in the core. In addition, a fraction of the radionuclides in a core cell can be designated as residing in the fuel-cladding gap.

Total radioactive class masses are normally determined by the DCH package from the operating power of the reactor and the mass of each element in the class per unit of operating power (see the DCH Package Reference Manual and Users' Guide). RN package input generally defines only the initial distribution of these masses in the core and cavity through reference values and multipliers specified on the RN1_FPN input records. However, options are provided to use these records to specify the class masses directly. These options are useful for analysis of experiments.

2.3 Release of Radionuclides

Release of radionuclides can occur from the core fuel (with nonradioactive releases from other core structures), from the fuel-cladding gap, and from material in the cavity. At present, no material can be released from the reactions treated in the FDI (fuel dispersal) package. The radionuclides residing in the COR package fuel are assumed to be in elemental form and, therefore, to have only radioactive mass (no associated molecular mass). Upon release from fuel, the total class masses are converted to compound form with a corresponding increase in mass from the added nonradioactive material (e.g., the hydroxide mass in CsOH). For core materials other than the fuel, such as the fuel rod cladding, the entire mass is nonradioactive.

Four options are currently available for the release of radionuclides from the core components; the CORSOR, CORSOR-M, CORSOR-BOOTH, or modified ORNL-BOOTH model may be specified on input record RN1_FP00. The CORSOR-BOOTH model contains low and high burn-up options. In addition, the CORSOR and CORSOR-M release rates can be modified to be a function of the component surface-to-volume ratio as compared to a base value, derived from the experimental data on which CORSOR is based. The reduction in release rate of the tellurium class by the presence of unoxidized zirconium is also modeled using the parameters in sensitivity coefficient array 7105.

By default the release models are used only to calculate the release of radioactive radionuclides from core fuel material (i.e. UO_2), which exists in the intact fuel component, in refrozen fuel material on other components, in particulate debris, and in the molten pool components. However, the same release correlations can be used to calculate the release of nonradioactive structural material from core components at their individual temperatures (e.g. Zr from the cladding as a function of the cladding temperature), if the user provides optional input to override the default (refer to sensitivity coefficient array 7100).

The release model also can provide for the combination of different donor classes into a new class based on the elemental molecular weights. An example could be the combination upon release of Cs and I atoms to form CsI molecules, which is modeled by moving stoichiometric amounts of Cs and I mass from the Cs and I classes into a new CsI class. The number of moles of each class that combine is defined by RN1_CLS input data. This combination occurs instantaneously upon release and is only limited by the availability of the released mass during that timestep. If there is an excess of any donor class during the timestep, that excess material stays in the original class.

It is assumed that the gaps in each radial ring can communicate axially between core cells, so when the cladding temperature in any core cell reaches or exceeds the cladding failure temperature specified for that cell, or when the cladding intact geometry has been lost, the entire gap inventory in that ring is released. A default temperature is provided, but may be overridden for any core cell using the RN1_GAP00 input record. This cladding failure temperature is only used in the RN package for gap releases and is not related to any COR package parameters.

For release of radionuclides from the cavity due to core-concrete interactions, the VANESA model has been implemented in MELCOR and is coupled to CORCON during every timestep. If a water pool is present, pool scrubbing calculations are performed to apportion the released mass between the pool and the atmosphere.

2.4 Aerosol Dynamics

The calculation of aerosol agglomeration and deposition processes is based on the MAEROS computer code. MAEROS is a multisectional, multicomponent aerosol dynamics code that evaluates the size distribution of each type of aerosol, or component, as a function of time. This size distribution is described by the mass in each size bin, or section, as depicted in Figure 2-1 of the RN Package Reference Manual. Each section may have a different chemical composition as described by the masses of various components for that section. In other words, a section is an aerosol size grouping and a component is a particular type of aerosol material.

Since MELCOR operates on a radionuclide class structure, a mapping between RN classes and MAEROS aerosol components must be specified by the user. The most accurate representation would be obtained with a one-to-one correspondence between classes and components. However, the computational cost of using 17 components in MAEROS can be high, and the increased accuracy is not thought to be justified, in general. The current default is to combine all 17 material classes into one MAEROS

components with water in a second component. The component representing only water droplets (class H2O aerosols) was added to improve the calculation of effects related to the condensation and evaporation of water, for only a small increment in resources. Note that calculations with additional components can be performed for comparison, if desired.

Aerosols can directly deposit onto heat structure and water pool surfaces through a number of processes, including gravitational settling, diffusion to surfaces, thermophoresis (a Brownian process causing migration of particles toward lower temperatures), and diffusiophoresis (deposition induced by condensation of water vapor onto structural surfaces). All heat structure surfaces are automatically designated as deposition surfaces for aerosols using information from the HS package. The surface orientation can be changed or deposition on a surface can be disabled through user input on the RN1_DS record.

Aerosols can also settle from one control volume to another through flow-through areas (i.e., the gravitational settling and Brownian diffusion kernels in MAEROS described below are applied to flow-through areas in addition to HS and pool surfaces). Such areas will ordinarily correspond to open flow paths between the control volumes, through which aerosols and radionuclide vapors are also advected. The appropriate settling area, path elevation, etc. are specified in the RN1_SET input record. Aerosols are not transported through these areas if the settling area is blocked by a water pool.

Finally, aerosols can agglomerate and become larger than the user-defined maximum diameter. These aerosols are assumed to immediately deposit onto water pools or horizontal heat structure surfaces or to settle from one control volume to another through flow-through *areas* defined as part of RN input. The term fallout in MELCOR is used exclusively for this immediate deposition or settling of aerosols larger than the maximum user-defined diameter. All control volumes must have at least one upward-facing deposition surface (floor) or flow-through area defined to receive fallout aerosols generated by this mechanism. If there is more than one, fallout is distributed in proportion to the total area of each surface. During MELGEN, a check is made for the existence of at least one such area; if none is present, an error message is generated, and no restart file is written.

A number of time-dependent aerosol sources (specified on tabular record RN1_AS) can also be specified for a control volume by the user (see the RN1_AS input tabular record rows). The aerosols can be put in either the control volume pool or atmosphere, with the time rate of the source specified by a tabular function.

2.5 Condensation/Evaporation and Hygroscopic Behavior

Fission products and water can condense onto or evaporate from aerosols, heat structure surfaces, and water pools. Aerosol water is identified with fog in the CVH package. The change in fog mass is determined by thermodynamics calculated within the CVH package and is then distributed over aerosol sections in the RN package by applying the Mason equation. Additionally, for water soluble aerosols, a solubility, or hygroscopic, effect is considered whereby the particles can grow by absorbing water

vapor from moist, unsaturated atmospheres. Water condensation and evaporation for heat structure and water pool surfaces are treated solely in the HS and CVH packages, respectively.

Neither condensation nor evaporation are included directly within the MAEROS solution framework, but are handled separately to reduce the stiffness of the differential equation set and to ensure consistency with the calculation of these processes by other models and packages. Instead, condensation and evaporation of fission product vapors to and from heat structures, pool surfaces, and aerosols are evaluated in RN with the same equations as in the TRAP-MELT2 code. The fission product vapor masses in the control volume atmosphere and condensed on the aerosol and heat structure surfaces are determined by rate equations based on the surface areas, mass transfer coefficients, atmosphere concentration, and the saturation concentrations corresponding to the temperatures of the surfaces.

2.6 Decay Heat Distribution

The decay heat released by radionuclides in the control volume atmosphere and from those deposited on the various heat structure surfaces can be apportioned according to user specifications on the appropriate RN_DHx records. The apportionment is among the volume atmosphere, the surfaces of heat structures in that volume, and the pool surface (if a pool is present). Fractions may also be specified as going to the atmosphere and surfaces of other volumes to simulate decay radiation transmitted through flow paths. All decay heat released by radionuclides in a control volume pool is assumed to be absorbed by that pool.

An approximate correction is made for the reduced deposition of decay heat in small or low density atmospheres when the thickness becomes comparable to or less than the range of typical beta radiation from fission product decay.

2.7 ESF Models

Models are currently available for the removal of radionuclides by pool scrubbing, filter trapping, and spray scrubbing. These models are controlled by the parameters input on the RN2_PLS, RN2_FCL, SPR_RN2_CLS, and SPR_RN2_HPART records (see the Containment Sprays [SPR] Package Users' Guide). The normal RN deposition and condensation models, including a surface area enhancement factor, are applied to heat structures used to model ice condensers (see the HS Package Users' Guide).

The pool scrubbing models, adapted from the SPARC90 code, include the effects of steam condensation at the pool entrance and aerosol deposition by Brownian diffusion, gravitational setting, and inertial impaction, subject also to evaporative forces, for the rising bubble. Decontamination is calculated only for those flow paths activated on the FL_JSW input record (see the FL Package Users' Guide). As further specified by the user on input record RN2_PLS, the model treats regular flow paths that vent through pools, as well as gases generated by core-concrete interactions flowing through overlying pools. Iodine vapor is also scrubbed. See Section 2.7.1 of the RN Package Reference Manual.

The MELCOR RN package contains a simple filter model. When aerosols and vapors are transported through flow paths with the bulk fluid flow of pool and/or atmosphere calculated by the CVH package, some fraction of the transported RN materials may be removed by the action of filters in the flow path. A single filter can remove either aerosols or fission product vapors, but not both. However, a flow path can contain more than one filter. The efficiency of each filter is defined by decontamination factors, specified by user input. By default, a single decontamination factor is applied to all RN classes except water, for which the default DF is 1.0. Additional user input may be used to modify the DF on a class-by-class basis, including the water class. The parameters for the filter characteristics are specified on the RN2_FLT input record.

The effect of filter mass loading on the flow resistance of the associated flow path may be modeled through user input. A maximum loading may be specified for each filter; when this loading is reached, no further RN materials will be removed (i.e., the DF is set to unity).

Several additional features are available to represent a variety of filter degradation and failure characteristics. These include radiolytic and thermal desorption of iodine vapor from charcoal filters, release of iodine from a charcoal filter due to charcoal combustion, and aerosol filter failure resulting from excessive mass loading. These models, developed for application of MELCOR to non-LWR plants, are not currently described in the RN Package Reference Manual.

The MELCOR SPR package, which calculates the thermal-hydraulic behavior associated with spray systems, is coupled to the RN package for the calculation of aerosol washout and atmosphere decontamination by the sprays. The spray model includes vapor adsorption and aerosol removal by diffusiophoresis, inertial interception and impaction, and Brownian diffusion. Aerosols and fission products removed by the sprays are deposited in the pool associated with the control volume or a user-defined sump pool.

2.8 Fission Product Chemistry

Chemistry effects can be simulated in MELCOR through the class reaction and class transfer processes using the RN1_RCT, RN1_RCTnn, RN1_TRN and RN1_TRNnn records. Reversible and irreversible reactions can be used to model adsorption, chemisorption, and chemical reactions. Only fission product vapors can react with surfaces and only vapors and ions produced from them can undergo chemical transformations in the pool.

2.9 Iodine Pool Chemistry Model

An iodine pool model has been implemented in MELCOR for use in predicting iodine in the containment atmosphere during the late phase of accident sequences. The model uses known iodine chemistry to predict what factors affect the iodine concentration in the atmosphere, while allowing for additional chemical reactions. In the containment atmosphere, where gas phase behavior is important, there are submodels relating the radiolysis of the air and cable insulation to the generation of nitric acid and hydrochloric

acid, respectively. On the structural surfaces, provision is made to account for the type of surface, thus allowing the extension to treat the effects of different paints and other surface coatings on iodine behavior. In the water pool, where liquid phase behavior is important, the model determines the pH based upon the user controlled boric acid and phosphate buffering, the effects of cesium hydroxide, cesium iodide and control rod silver released by the accident scenario chosen, and the effects of the acids introduced from the containment atmosphere due to radiolysis. The aqueous pool chemistry model then determines the speciation of iodine, particularly the important elemental, molecular, and organic forms, over the range of pH from 4 to 12. Thus, chemical systems that control pool pH can be examined as well as pools and films on surfaces that have no pH controls. With this combination of features, the iodine pool model provides the ability to conduct sensitivity studies and to incorporate new effects found in the course of ongoing research.

3 User Input

3.1 MELGEN Input

The input description for the RN package for MELGEN follows. As noted occasionally below, several input parameters define models or options not used with the LWR version of MELCOR, and which are not described in the LWR COR or RN reference manuals. CAUTION: If the name of the radionuclide class is changed (different than the default name), user must check MELGEN output to ensure that class mapping is done correctly between VANESA class to RN class, and vice versa. In most cases, the user should add RN1_VNCL and RN1_CLVN cards to ensure the correct class mapping.

3.1.1 General Control, Options, and Mappings

These input records define various dimensions, options, and mappings that control the framework and structure of the models in the RN Package.

RN1_INPUT – Start record for RN1 package user-input

Required.

This record activates the RN package in MELCOR. When the RN1_INPUT record is absent, then by default, the activation switch is set to not active.

(1) IACTV

Activation switch for the RN package. Optional field.

(a) 0 or ACTIVE
RN package Active

(b) 1 or NOTACTIVE
RN package Not Active

(type = integer / character*9, default=0 (active), units = none)

Example

```
RN1_INPUT ACTIVE
```

RN1_DFT – RN Package Default Scheme

Optional.

This record modifies a set of RN package sensitivity coefficient values to those 1.8.6 standards. Note that these values may not reflect current MELCOR best practice modeling parameters and sensitivity coefficients for the analysis of severe accidents. If the record is present, one floating point field must be present.

(1) DEFAULT

Mode of default scheme. Permitted values are 1.86 and 2.0.

(type =real, default=2.0, units = none)

If the DEFAULT field is 2.0, no change is made to the existing sensitivity coefficients. If the DEFAULT field is 1.86, the following RN package sensitivity coefficients will be modified, unless otherwise specified by the user input record. Note that the following table only contains rows for those Classes that are changed between 1.86 and 2.0.

Class	**Sensitivity Coefficient**	**DEFAULT = 1.86**	**DEFAULT = 2.0**
CORSOR-Booth Class Scaling Factors: Nominal Values			
BA	7103	3.33E-03	4.0E-04
I2	7103	1.0	6.4E-01
TE	7103	1.0	6.4E-01
RU	7103	1.0E-04	2.5E-03
CE	7103	3.34E-05	4.0E-08
LA	7103	1.0E-04	4.0E-08
UO2	7103	1.0E-04	3.2E-04
CD	7103	5.0E-02	2.5E-01
AG	7103	5.0E-02	1.6E-01
CSI	7103	0.0	6.4E-01
CSM	7103	0.0	1.0
CORSOR-Booth Transient Release Parameters for Cs			
	7106 (1, 1)	5.0E-08	2.3E-09
	7106 (2, 1)	2.5E-07	2.3E-09
	7106 (4, 1)	3.8E+05	2.411E+05
	7106 (5, 1)	1.0E-05	6.0E-06
CORSOR-Booth Class Scaling Factors: Oxidation Modified			
BA	7107 (6)	5.0E-02	5.0E-01
	7107 (7)	5.0E-02	2.0E-03
TE	7107 (7)	2.5E-02	6.4E-01

Class	Sensitivity Coefficient	DEFAULT = 1.86	DEFAULT = 2.0
RU	7107 (3)	1.06792E-20	2.5E-03
	7107 (4)	1.59923E-02	0.0
LA	7107 (7)	1.0E-01	4.0E-08
CD	7107 (3)	3.194E-09	2.5E-01
	7107 (4)	8.283E-03	0.0
AG	7107 (3)	3.194E-09	1.6E-01
	7107 (4)	8.283E-03	0.0
Vapor Pressures			
CS	7110 (1, 2)	9.4E+03	No Change
	7110 (1, 3)	2.159E+01	No Change
	7110 (1, 4)	-3.75	No Change*
	7110 (2, 1)	1.553E+03	No Change*
	7110 (2, 2)	6.870778E+03	No Change*
	7110 (2, 3)	7.994503	No Change*
BA	7110 (1, 2, 3)	7.836E+03	No Change*
	7110 (1, 3, 3)	6.44	8.4
	7110 (2, 1, 3)	-1.0	1.0E+04
RU	7110 (1, 3, 6)	1.06088E+01	1.16088E+01
	7110 (2, 1, 6)	-1.0	1.0E+04
CSM	7110(1,1,17)	(Not Used)	600.0
CSM	7110(1,2,17)	(Not Used)	13600.0
	7110(1,3,17)	(Not Used)	8.895
	7110(1,4,17)	(Not Used)	0.0
	7110(2,1,17)	(Not Used)	1229.5
	7110(2,2,17)	(Not Used)	12100.0
	7110(2,3,17)	(Not Used)	7.675
	7110(2,4,17)	(Not Used)	0.0
Hygroscopic Aerosol Parameters			
CSM	7170(3,17)	(Not Used)	0.67
	7170(4,17)	(Not Used)	0.67
	7170(9,17)	(Not Used)	4030.0

*Note that the default for Cs has not changed though for earlier versions of MELCOR 2.X, this was not the case.

Example

```
RN1_DFT    1.86
```

RN1_DIM – Dimension

Optional.

This record defines the dimensions for the database. If this record appears then the first three fields are required, fourth field is optional.

(1) NUMSEC

Number of sections in the aerosol calculation.

(type = integer, default = 10, units = none)

(2) NUMCMP

Number of aerosol components.

(type = integer, default = 2, units = none)

(3) NUMCLS

Number of material classes, must be the same as the number of classes specified in the DCH input.

(type = integer, default = 17, units = none)

(4) NUMCA

Number of chemisorption classes. Optional field.

(type = integer, default = 6, units = none)

Example

```
!        NUMSEC  NUMCMP  NUMCLS  NUMCA
RN1_DIM  15      17      2
```

RN1_HM – Activate Hygroscopic Model

Optional.

This record activates the hygoscopic model in MELCOR.

(1) IHYGRO

Activation switch for hygroscopic model.

(a) 0 or NOTACTIVE
Hygroscopic model off.

(b) 1 or ACTIVE
Hygroscopic model on.

(type = integer / character*9, default = 0 (not active), units = none)

Example

```
RN1_HM  NOTACTIVE  ! hygroscopic model off
```

RN1_DCHNORM - Class Normalization

Optional.

This record allows specification of the RN classes for which the mass (kg) and decay heat (W/kg) in the DCH reference core is to be modified to reconcile the total mass with the initial mass defined by user input while preserving the total decay heat associated with the class. This is ordinarily used only for the URANIUM class, where the initial mass is derived from input on COR_KFU records, and may differ from the built-in reference core. The masses of all other classes are specified by RN1_FPN input, usually in terms of fractions of the total, so that there should normally be no discrepancy. The default is to perform this reconciliation for class URANIUM (10 by default) only. There are three forms of input

(1) ICLNRM

Keyword ALL to indicate that all classes are to be reconciled.

(type = character*3, default = none, units = none)

or

(1) ICLNRM1

Keyword NONE to indicate that no classes are to be reconciled.

(type = character*4, default = none, units = none)

or

(1) NNORM

Length of Class Normalization Table.

(type = integer, default = 0, units = none)

And, the following data are input as a table with length NNORM:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLNRM

Class name for which DCH reference core is to be reconciled with RN input. Each name must be defined as a radionuclide class in the DCH package, but there is no check for duplicated class names.

(type = character*16, default = none, units = none)

Example

```
RN1_DCHNORM   NONE   ! No reconciliation
```

RN1_CRCL – Core Material to RN Class Map

Optional.

The mapping of the nonradioactive core mass to the RN material class structure is determined by this input. This mapping is discussed in more detail in the model description section of the reference manual.

(1) NMCRCL

Number of data strings in the Core Material to RN Class table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMCRCL:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) ICRMT

Core material:

(a) 1 or FUELMAT
Fuel material in COR package

(b) 2 or UNOXZR
Unoxidized Zirconium in COR package

(c) 3 or OXZR
Oxidized Zirconium in COR package

(d) 4 or UNOXST
Control rod unoxidized steel in COR package

(e) 5 or OXST
Control rod oxidized steel in COR package

(f) 6 or POISON
Control rod poison in COR package

(type = integer / character*7, default = see below, units = none)

(3) ICLSS

RN material class name (see Table 1 or any user-defined in the DCH input class name).

(type = character*16, default = see below, units = none)

(4) FRAC

Fraction of the core material, ICRMT, that is in class ICLSS.

(type = real, default = see below, units = none)

Default values are:

ICRMT	ICLSS	Fraction
1	UO2	1.0
2	Ce	1.0
3	Ce	1.0
4	Mo	1.0
5	Mo	1.0
6	BO2	1.0 (for BWRs)
6	Cd	0.05 (for PWRs)
6	Ag	0.95 (for PWRs)

The default values from the table above will be used only for those classes defined by the user.

Example

```
RN1_CRCL 1 !NSTR Core Material  RN material  Fraction
             1      POISON         BO2          1.0
```

RN1_CLVN – RN Class to VANESA Group Map

Optional.

When debris enters the cavity, the associated radionuclides are converted to the VANESA group structure and maintained in that form until released from the melt. The mapping of the RN class masses to the masses in the VANESA structure is determined by this input. This input was provided for maximum flexibility to allow a user to redefine RN classes. Input should be consistent with the RN class structure; non-default values should be used with great care, if at all. The complete list of VANESA group numbers is described in the RN Package Reference Manual.

(1) NMCLVN

Number of data strings in the RN Class to the VANESA Group table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMCLVN:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) ICLSS

RN material class name.

(type =character*16, default = see below, units = none)

(3) ICLSSV

VANESA Group Number (only numbers from 1 to 27 are available).

(type = integer, default = see below, units = none)

Default values are:

ICLSS	ICLSSV
Xe	27 (Xe, released instantaneously)
Cs	19 (Cs)
Ba	20 (Ba)
I2	26 (I, immediately forms CsI)
Te	9 (Te)
Ru	6 (Ru)
Mo	5 (Mo)
Ce	23 (Ce)
La	22 (La)
UO2	17 (U)
Cd	8 (Sb)
Ag	7 (Sn)
BO2	(RN class not present in fuel)
H2O	(RN class not present in fuel)
Con	(RN class not present in fuel)
CsI	25
CsM	19(Cs) and 5(Mo)

Note that in mapping CsM to Cs and Mo, the inventory of Cs and Mo may increase as they may be released from Vanesa due to MCCI interactions.

Masses for RN classes BO2, H2O, and Con are not present in the fuel, so no mapping is required for these classes.

The default values from the table above will be used only for those classes defined by the user.

Example

```
RN1_CLVN 1 ! NSTR  ICLSS  ICLSSV
              1     CSI    25
```

RN1_VNCL – VANESA Group to RN Class Map

Optional.

On release, VANESA group masses are converted to RN class form. The mapping of the VANESA group masses to the RN class masses is input on this record. This input was provided for maximum flexibility to allow a user to redefine RN classes.

Input should be consistent with the RN class structure; non-default values should be used with great care, if at all. The complete list of VANESA group numbers is described in the RN Package Reference Manual.

(1) NMVNCL

Number of data strings in the VANESA Group to RN Class table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMVNCL:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) ICLSSV

VANESA Group Number (only numbers from 1 to 27 are available).

(type = integer, default = see below, units = none)

(2) ICLSS

RN material class name.

(type = character*16, default = see below, units = none)

Default values are:

ICLSSV	ICLSS	ICLSSV	ICLSS
1 (gases)	none	14 (Na)	CON
2 (Fe)	Mo	15 (K)	CON
3 (Cr)	Mo	16 (Si)	CON
4 (Ni)	Ru	17 (U)	UO2
5 (Mo)	Mo	18 (Zr)	Ce
6 (Ru)	Ru	19 (Cs)	Cs
7 (Sn)	Ag	20 (Ba)	Ba
8 (Sb)	Cd	21 (Sr)	Ba
9 (Te)	Te	22 (La)	La
10 (Ag)	Ag	23 (Ce)	Ce
11 (Mn)	Mo	24 (Nb)	Mo
12 (Ca)	CON	25 (CsI)	CsI
13 (Al)	CON	26 (I)	None
		27 (Xe)	None

CsI (group 25) receives special treatment, as described in the RN Package Reference Manual. Bulk gases from CORCON (group 1) are released by the CAV package to the CVH package. I (group 26) is automatically combined with Cs by

VANESA, and Xe (group 27) is released instantaneously by VANESA; no mapping is needed for these groups.

The default values from the table above will be used only for those classes defined by the user.

Example

```
RN1_VNCL  1 ! Number   Vanesa Group Number    RN Class Name
                1              25               'CSI'
```

RN1_DIFMAP - Maps tracked HTGR species to the RN classes

Optional.

(1) NN

Number of table lines.

(type = integer, default = none, units = none)

The following are input as a table with NN lines:

(1) N

Line index.

(type = integer, default = none, units = none)

(2) NCLS

RN class name

(type = character*16, default = none, units = none)

(3) JSPEC

Tracked species controlling RN class NCLS

(type = character*16, default = none, units = none)

(4) ASSOC

Flag indicating whether this is the assigned RN class for species JSPEC.

(a) Y

This is the assigned RN class.

(b) N

This is not the assigned class.

(type = character*16, default = none, units = none)

(5) SCALE

Scale factor to multiply species release, used similarly to scaling factor for Cs release in present Corsor-Booth model.

(type = real, default = 1.0, units = none)

Example

```
RN1_DIFMAP  1 ! Number   RN ClassName FP SpeciesName Assoc Scale
                1        CS           'Cs'           N     1.0
```

3.1.2 Initial Radionuclide Inventories

The initial inventories of radionuclides in the core, cavity, and control volumes are given by these records.

Note that these data determine the amount of decay heat in the core and cavity and, when released, the decay heat in the aerosols produced from the core and cavity. If these data are not input, <u>there will be no decay heat in the problem from material released from the core or cavity</u>. Decay heat will be generated for the initial radionuclide masses and from sources depending on the radioactive fractions specified. If the RN package is not active, the total decay heat value calculated by the DCH package will be split between the core and the cavity according to the amount of fuel in each location.

RN1_FPN – Initial Core Fuel and Cavity Radionuclide Inventories

Optional.

Each data string specifies the fission product mass in a given location or the multiplier on the total mass for each class. The total mass is the mass used in the DCH package for each class except for class uranium, which is taken from the COR package database (i.e., user-defined fuel distribution).

(1) NMFPN

Number of data strings in the Initial Core Fuel and Cavity Radionuclide Inventories table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMFPN:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) IR

Depends on COR or CAV input:

(a) > 0

Then IR is the core radial node number and the axial node number is input on field 3.

(b) = 0
Then cavity input and the user-defined cavity name is input on field 3.

(type = integer, default = none, units = none)

optionally followed by

(3) IA
The core axial node number.

(type = integer, default = none, units = none)

or

(3) CAVNAM
The user-defined cavity name.

(type = character*16, default = none, units = none)

(4) NINP
Flag to determine type of input:

(a) -1 or CELL
Then the reference node for the radionuclide inventory in the present node must be input on fields IRREF and IAREF. This value of NINP is only possible for IR > 0.

(b) 0 or DH
Use the total mass of each class as given in DCH or COR package input.

(c) 1 or CLASS
Then the class name for the radionuclide mass must be input on field CLSNAM.

(type = integer / character*5, default = none, units = none)

optionally followed by

(5) IRREF
The core radial ring number for the reference node.

(type = integer, default = none, units = none)

and

(6) IAREF
The core axial segment number for the reference node.

(type = integer, default = none, units = none)

or

(5) CLSNAM

The class for the radionuclide mass read on the next entry.

(type = character*16, default = none, units = none)

The last two fields fifth and sixth if NINP = 0 or DH, sixth and seventh if NINP = 1 or CLASS, seventh and eighth if NINP = -1 or CELL.

(n) RINP1

Variable which depends on value of NINP.

If NINP = CELL or -1 or NINP = DH or 0, multiplier on this reference inventory for the present node, for example, an axial node multiplier.

If NINP = CLASS or NINP = 1, Initial radionuclide mass (in kg) of class NINP.

(type = real, default = none, units = none or kg)

(n+1) RINP2

Multiplier on this reference inventory for the present node, for example, a radial node multiplier.

If NINP = CELL or -1 or NINP = DH or 0,
Mass= Mass (defined by NINP value) x RINP1 x RINP2

If NINP = CLASS or NINP = 1,
Mass= Mass (RINP1 value) x RINP2

(type = real, default = none, units = none)

The information on these records is additive.

Examples

```
RN1_FPN 5 !N   IR IA  NINP  IR,IA/CLSNAM      RINP1     RINP2
          1    1  1  DH                      0.154    0.1656
          2    1  2  CELL   1  1             1.273    1.0
          3    1  3  CELL   1  1             1.29     1.0
          4    1  4  CELL   1  1             1.319    1.0
          5    1  20 CELL   1  1             1.319    1.0
RN1_FPN 8 !N IR CAVNAM/IA NINP IR,IA/CLSNAM RINP1    RINP2
          1  1         4  DH               0.2      0.6
          2  1         5  DH               0.4      0.6
          3  1         6  DH               0.4      0.6
          4  1         7  DH               0.2      0.6
          5  2         4  DH               0.2      0.4
          6  2         5  DH               0.4      0.4
          7  2         6  DH               0.4      0.4
          8  2         7  DH               0.2      0.4
```

RN1_GAP – Initial Fuel-Cladding Gap Inventory Fractions

Optional.

The amount of core radionuclide material in the gap region is specified on this record. Note that the total amount of radionuclides in the fuel and the gap is given on the RN1_FPN record. The present record is used to distribute the total amount between the fuel and the gap. Upon cladding failure, the gap inventory of the entire radial ring is released to the appropriate control volume. In addition, any release of radionuclides from the fuel is held up in the gap until cladding failure. Therefore, a puff-type release is usually seen when the cladding fails.

(1) NMGAP

Number of data strings in the Initial Fuel-Cladding Gap Inventory Fractions table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMGAP:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) IR

The core radial node number.

(type = integer, default = none, units = none)

(3) IA

The core axial node number.

(type = integer, default = none, units = none)

(4) NINP

Flag to determine type of input:

(a) -1 or CELL

Then the reference node for the gap inventory fractions in the present node must be input on fields IRREF and IAREF.

(b) 1 or CLASS

Then the class for the gap fraction must be input on field CLSNAM.

(type = integer / character*5, default = none, units = none)

optionally followed by

(5) IRREF

The core radial ring number for the reference node.

(type = integer, default = none, units = none)

and

(6) IAREF

The core axial level number for the reference node.

(type = integer, default = none, units = none)

or

(5) CLSNAM

The class for the gap fraction input on the next entry.

(type = character*16, default = none, units = none)

followed by

(n) RINP1

Variable which depends on value of NINP.

If NINP = CELL or NINP = -1, Multiplier on the fraction of inventory in the gap of the reference node for use in the current node.

If NINP = CLASS or NINP = 1, fraction of total node inventory of class CLSNAM that is in the gap.

(type = real, default = none, units = none)

(n+1) RINP2

Ratio of total mass to radioactive mass, required if and only if NINP = 1 or CLASS.

(type = real, default = none, units = none)

Example

```
RN1_GAP  12 !N  IR  IA  NINP  IR,IA/CLSNAM    RINP1   RINP2
             1   1  4   CLASS       CS        0.05      1.0
             2   1  4   CLASS       I2        0.017     1.0
             3   1  4   CLASS       XE        0.03      1.0
             4   1  4   CLASS       TE        0.0001    1.0
             5   1  4   CLASS       BA        0.000001  1.0
             6   1  5   CELL   1  3           1.0
             7   1  6   CELL   1  3           1.0
             8   1  7   CELL   1  3           1.0
             9   2  4   CELL   1  3           1.0
            10   2  5   CELL   1  3           1.0
            11   2  6   CELL   1  3           1.0
            12   2  7   CELL   1  3           1.0
```

RN1_AG – Initial Aerosol Masses in Atmosphere

Optional.

This record allows the user to input an initial mass of aerosol in the gas phase (atmosphere) of any CVH control volume.

(1) NAME

User-defined name for aerosol mass table.

(type = character*16, default = none, units = none)

(2) IVOL

User-defined name of the control volume containing initial aerosol masses.

(type = character*16, default = none, units = none)

(3) ICLSS

Class name of input aerosol masses. An initial mass of water aerosol (H2O class [14 by default]) is permitted if and only if the hygroscopic aerosol modeling option has been selected.

(type = character*16, default = none, units = none)

(4) RFRAC

Radioactive fraction of the aerosol masses.

(type = real, default = none, units = none)

Initial aerosol masses of class ICLSS in sections in atmosphere of control volume IVOL are specified in the following table. (The length of the table must be equal to NUMSEC.)

(1) N

Section index.

(type = integer, default = none, units = none)

(2) XMASS(N)

Initial aerosol mass of class ICLSS in section N in atmosphere of control volume IVOL.

(type = real, default = none, units = kg)

A total of NUMSEC values are required. Input for the same control volume and class is additive.

Example

```
!      NAME      IVOL              ICLSS  RFRAC
RN1_AG Aerosol1  CV150              BA     1.0
!N     XMASS
```

```
 1     0.00001
 2     0.00001
 3     0.00002
 4     0.00001
!      NAME      IVOL            ICLSS   RFRAC
RN1_AG Aerosol2  CV150             Fe     1.0
!N     XMASS
 1     0.00001
 2     0.00001
 3     0.00002
 4     0.00001
```

RN1_ADFG – Aerosol Deposition Model Deactivation Flag

Optional.

This record allows the user to input flags that enable the aerosol deposition models to be turned off.

(1) OFF_ADEP

Keywords to turn off the specific deposition models. Up to three keywords can be entered. See below the lists of keywords for the specific model.

(a) "GRV_OFF"
Deactivate gravitational deposition model

(b) "DIF_OFF"
Turn off the diffusive deposition model

(c) "THM_OFF"
Turn off the thermophoresis deposition model

(type = character*7, default = none, units = none)

Example

```
RN1_ADFG   GRV_OFF DIF_OFF THM_OFF ! turn off all depositions
```

RN1_AL – Initial Aerosol Masses in Pool

Optional.

This record allows the user to input an initial mass of aerosol in the liquid phase (pool) of any volume.

(1) NMALIN

Number of data strings in the Initial aerosol masses in pool table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMALIN:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) IVOL
User volume name for input masses.
(type = character*16, default = none, units = none)

(3) ICLSS
Class name of input aerosol masses.
(type = character*16, default = none, units = none)

(4) RFRAC
Radioactive fraction of masses.
(type = real, default = none, units = none)

(5) XMASS
Initial aerosol mass of class ICLSS in pool of volume IVOL.
(type = real, default = none, units = kg)

Input for the same control volume and class is additive.

Example

```
RN1_AL 1 ! NSTR  IVOL     ICLSS  RFRAC   XMASS
             1   CV110     CSI    1.0     0.02
```

RN1_VG – Initial Fission Product Vapor Masses in Atmosphere

Optional.

Initial fission product vapor masses in the atmosphere can be input on this record.

(1) NMVGIN
Number of data strings in the initial fission product vapor masses in atmosphere table.
(type = integer, default = 0, units = none)

The following data are input as a table with length NMVGIN:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) IVOL

User volume name for input masses.

(type = character*16, default = none, units = none)

(3) ICLSS

Class name of input aerosol masses.

(type = character*16, default = none, units = none)

(4) RFRAC

Radioactive fraction of masses.

(type = real, default = none, units = none)

(5) XMASS

Initial vapor mass of class ICLSS in atmosphere of volume IVOL.

(type = real, default = none, units = kg)

Input for the same control volume is additive.

Example

```
RN1_VG 3 !NSTR IVOL      ICLSS  RFRAC    XMASS
           1   CV150      XE     1.0      0.0
           2   CV150      CS     1.0      0.1
           3   CV150      BA     1.0      0.0
```

RN1_VL – Initial Fission Product Vapor Masses in Pool

Optional.

Initial fission product vapors in the pool can be input through this record.

(1) NMVLIN

Number of data strings in the initial fission product vapor masses in pool table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMVLIN:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IVOL

User volume name for input masses.

(type = character*16, default = none, units = none)

(3) ICLSS

Class name of input vapor masses.

(type = character*16, default = none, units = none)

(4) RFRAC

Radioactive fraction of masses.

(type = real, default = none, units = none)

(5) XMASS

Initial vapor mass of class ICLSS in pool of volume IVOL.

(type = real, default = none, units = kg)

Input for the same control volume is additive.

Example

```
RN1_VL 3 ! NSTR   IVOL   ICLSS  RFRAC   XMASS
            1    CV150     XE    1.0     0.0
            2    CV150     CS    1.0     0.1
            3    CV150     BA    1.0     0.0
```

RN1_DSCAL - Transport Deposition Scaling

Optional.

This is used to set the scaling for the RN classes initialized in the TRANSFILE. The classes are usually scaled differently: some are constant amounts in the CVs, some increase with time, some have a deposition rate. The class scaling is applied globally, i.e., all HSs get the same class scaling, etc.

(1) NN

Number of table lines.

(type = integer, default = none, units = none)

The following are input as NN table lines:

(1) N

Line index.

(type = integer, default = none, units = none)

(2) CLASS

RN class name.

(type = character*16, default = none, units = none)

(3) OBJ

Flag for object type, HSs, CVs, or filters.

(a) HS

Heat Structures.

(b) CV
Control Volumes.

(c) FILT
Filters.

(type = character*16, default = none, units = none)

(4) AMTRATE

Flag as to whether amount or rate is to be scaled.

(d) AMT
Scale amount on/in OBJ.

(e) RATE
Scale rate of deposition on/in OBJ.

(type = character*16, default = none, units = none)

(5) SCALE

Scale factor that overrides the GSCALE on the COR_DIFT record for this class on this OBJ.

(type = real, default = 1.0, units = none)

3.1.3 Release Model Parameters

RN1_FP00 – Core Release Model

Optional.

(1) ICRLSE

Release model indicator for the core.

(type = integer, default = -5, units = none)

For oxide fuels, ICRLSE is interpreted as follows:

(a) -1 or C_SVO
Original CORSOR model with surface-to-volume ratio option.

(b) 1 or CORS
Original CORSOR model without surface-to-volume ratio option.

(c) -2 or CM_SVO
CORSOR-M model with surface-to-volume ratio option.

(d) 2 or CM

CORSOR-M model without surface-to-volume ration option.

(e) -3 or CB_HBF
CORSOR-Booth model for high burn-up fuel.

(f) 3 or CB_LBF
CORSOR-Booth model for low burn-up fuel.

(g) -5 or RCB_HBF
Revised CORSOR-Booth model for high burn-up fuel.

(h) 5 or RCB_LBF
Revised CORSOR-Booth model for low burn-up fuel.

(i) 6 or GENREL or GENERALRELEASE or BURST
Generalized release model.

Note: ICRLSE=±4 is also accepted as input, but should not be used. It specifies use of an undocumented experimental release model.

Warning: The default coefficients used in all of the CORSOR models are for the 16 default material classes. For material class definitions other than the default specification, the user may need to alter the CORSOR models through sensitivity coefficients (7101 through 7107). The corresponding vapor pressure (7110), vapor diffusivity (7111) and class molecular weights (7120) sensitivity coefficient arrays may also need to be altered for consistent release modeling. DCH package input must also be modified in a consistent manner.

Example

```
RN1_FP00 'C_SVO'
```

RN1_BOOTH00 – Identification of Cs class(es) for the CORSOR-Booth release model

Optional.

This record identifies the name of the RN class that is considered as Cs. If the second field is present, it is used to identify the Cs class for release from an alternate fuel.

(1) ICLCSBO
Name of Cs class for purposes of the CORSOR-Booth release model.

(type = character*16, default = 'CS', units = none)

(2) ICLCSBOA
Name of Cs class for purposes of application of the CORSOR-Booth release model to an alternate fuel type.

(type = character*16, default = ' ', units = none)

Example

```
RN1_BOOTH00 CS   BA
```

RN1_BOOTH01 – Identification of classes associated with alternate fuel

Optional.

This record must contain the class name ICLCSBOA from the previous record (if defined) in order for the leading class in this group to belong to alternate fuel.

(1) ICLALT

A number of RN classes associated with alternate fuel for purposes of the CORSOR-Booth release model, the dimension of the following table.

(type = integer, default = 0, units = none)

The following data are input as a table with length ICLALT:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NAME

Name of an RN class associated with alternate fuel.

(type = character*16, default = none, units = none)

Example

```
RN1_BOOTH01 1  ! N     Class name
                 1        BA
```

RN1_GREL – Class Multipliers for General Release Model

Optional.

This record must contain the class name ICLCSBOA from the previous record (if defined) in order for the leading class in this group to belong to alternate fuel.

(1) ICLALT

A number of RN classes associated with alternate fuel for purposes of the CORSOR-Booth release model, the dimension of the following table.

(type = integer, default = 0, units = none)

The following data are input as a table with length ICLALT:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NAME

Name of an RN class associated with alternate fuel.

(type = character*16, default = none, units = none)

Example

```
RN1_BOOTH01 1  ! N     Class name
                 1        BA
```

RN1_DIFF – Class Parameters for Diffusion Release Fraction

Optional.

This record must contain the class name ICLCSBOA from the previous record (if defined) in order for the leading class in this group to belong to alternate fuel.

(1) ICLALT

A number of RN classes associated with alternate fuel for purposes of the CORSOR-Booth release model, the dimension of the following table.

(type = integer, default = 0, units = none)

The following data are input as a table with length ICLALT:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NAME

Name of an RN class associated with alternate fuel.

(type = character*16, default = none, units = none)

Example

```
RN1_BOOTH01 1  ! N     Class name
                 1        BA
```

RN1_FP00 –

Optional.

This record must contain the class name ICLCSBOA from the previous record (if defined) in order for the leading class in this group to belong to alternate fuel.

(1) ICLALT

A number of RN classes associated with alternate fuel for purposes of the CORSOR-Booth release model, the dimension of the following table.

(type = integer, default = 0, units = none)

The following data are input as a table with length ICLALT:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NAME

Name of an RN class associated with alternate fuel.

(type = character*16, default = none, units = none)

Example

```
RN1_BOOTH01 1  ! N     Class name
                 1        BA
```

RN1_GAP00 – Gap Release Temperature

Optional.

(1) NMGAP0

Number of data strings in the gap release temperature table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NMGAP0:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IR

The core radial node number.

(type = integer, default = none, units = none)

(3) IA

The core axial node number.

(type = integer, default = none, units = none)

(4) CLFAIL

Cladding failure temperature for gap release for core cell (IA, IR).

If CLFAIL = -1, then the cladding does not fail until the core cell goes from intact geometry to debris geometry

(type = real, default=1173., units = K)

Example

```
RN1_GAP00   8   ! N   IR  IA        Failure Temperature
                  1    1   4             1173.0
                  2    1   5             1173.0
                  3    1   6             1173.0
                  4    1   7             1173.0
                  5    2   4             1173.0
                  6    2   5             1173.0
                  7    2   6             1173.0
                  8    2   7             1173.0
```

RN1_CLS – Parameters for Class Combination at Release

Optional.

This record specifies the combination of masses in one or more donor classes into a new acceptor class, e.g., masses from the Cs and I classes into the CsI class. This combination is only applied to masses at initial release from core materials.

(1) CCNAME

User-defined name for class combination.

(type = character*16, default = none, units = none)

(2) NCLCN

Acceptor class name.

(type = character*16, default = none, units = none)

(3) NCLCT

The number of donor classes for class NCLCN.

(type = integer, default = none, units = none)

The following data are input as a table with length NCLCT:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NCLSD

Donor class name.

(type = character*16, default = none, units = none)

(3) XMRAT

Ratio of the number of moles transferred from the donor class to the number of moles received by the acceptor class (e.g., 1.0 for Cs or I to CsI).

(type = real, default = none, units = none)

Example

```
RN1_CLS ClassComb1 CSI 2 ! NSTR DONOR CLASS   XMRAT
                             1      CS         1.0
                             2      I2         0.5
```

3.1.4 RN Sources

RN1_AS – Aerosol Source

Optional.

This record must be input for each aerosol source, if any exists. A source of aerosol mass can be input with this record, defined by a tabular or control function. If the source is directed to the liquid phase (pool) and none is present, the source will be accumulated and added to the pool if it reappears.

(1) NAME

User-defined aerosol source name.

(type = character*16, default = none, units = none)

(2) IVOL

User volume for aerosol source.

(type = character*16, default = none, units = none)

(3) IPHS

Phase to receive aerosol source mass.

(a) 1 or LIQUID
Liquid phase (pool).

(b) 2 or VAPOR
Vapor phase (atmosphere).

(type = integer / character*6, default = none, units = none)

(4) ICLSS

Class name of aerosol source.

(type = character*16, default = none, units = none)

(5) RFRS

Radioactive fraction of mass source.

(type = real, default = none, units = none)

(6) XM

Mass addition rate, multiplies the value of tabular or control function specified by ITAB below.

(type = real, default = none, units = kg/s)

(7) ITAB

Definition of the time dependence of the mass addition rate XM.

(a) 1 or TF
Use tabular function specified by CFTFNAME below.

(b) 2 or CF
Use control function specified by CFTFNAME below.

(type = integer / character*2, default = none, units = none)

(8) CFTFNAME

Name of tabular or control function defining the time dependence of the mass addition rate.

(type = character*16, default = none, units = none)

Example

```
RN1_AS   ASR1   CV1   VAPOR   CS   0.0  9.5E-4 TF TF5
RN1_AS 'COR SRC' CV50 LIQUID TE 0.1 240.5 CF CF505
RN1_AS 'CAV SRC' CV50 LIQUID MO 0.22 1.85 TF TF507
RN1_AS 'SECONDARY SRC' CV50 VAPOR  I2 0.02 1.0 CF CF515
```

In the examples, the radioactive fraction RFRS includes the ratio of (element MW)/(compound MW), as the aerosol source mass goes into the total class mass array. The radioactive class mass is then set from the total mass x RFRS.

RN1_AS01 – Sectional Distribution Parameter

Required if record RN1_AS appears and IPHS on this record equals 2 or VAPOR.

This record is additional to RN1_AS record and it is only necessary for atmosphere sources.

(1) IDIST

Sectional distribution parameter, only necessary for atmosphere sources (IPHS = 2 or VAPOR on record RN1_AS).

(a) 1 or UNIFORM
Uniform source with respect to log diameter (i.e., log uniform), no additional values.

(b) 2 or LOGNORMAL
Log-normal distribution with respect to log diameter. Two additional values described below, GEOMM (2) and GSD (3), are needed, the data are input on this data string.

(c) 3 or SECTIONBYSECTION
Section-by-section distribution specified. NUMSEC additional values for FRC below are needed, the data are input from the new line as the table.

(type = integer / character*16, default = none, units = none)

If IDIST = 2 or IDIST = LOGNORMAL then

(2) GEOMM

Source mean diameter (also denoted AMMD). If greater than zero the value is the mass median diameter. If less than zero, the absolute value is the geometric mean diameter.

(type = real, default = none, units = m)

(3) GSD

Source geometric standard deviation.

(type = real, default = none, units = none)

If IDIST = 3 or IDIST = SECTIONBYSECTION then the table with length NUMSEC is input from the new line:

(1) NS

Section index.

(type = integer, default = none, units = none)

(2) FRC

Fraction of source in each section. Values are normalized after they are read.

(type = real, default = 1.0, units = none)

Example

```
RN1_AS01  SECTIONBYSECTION ! NS   FRACTION OF SOURCE
                             1            0.5
                             2            1.0
                             3            1.0
                             4            1.0
                             5            1.0
```

RN1_VS – Vapor Source

Optional.

Sources of fission product vapors are input on this record.

(1) NUMSRV

Number of the tabular vapor sources, the length of the vapor sources table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMSRV:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IVOL

User volume name for vapor source.

(type = character*16, default = none, units = none)

(3) IPHS

Phase for vapor source.

(a) 1 or LIQUID
Liquid phase.

(b) 2 or GAS
Gas phase.

(type = integer / character*6, default = none, units = none)

(4) ICLSS

Name of class of vapor source.

(type = character*16, default = none, units = none)

(5) RFRC

Radioactive fraction of mass source.

(type = real, default = none, units = none)

(6) XM

Mass addition rate. Time dependence is included through the tabular or control function specified by ITAB and CFTFNAME below.

(type = real, default = none, units = none)

(7) ITAB

Definition of the time dependence of the mass addition rate XM.

(a) 1 or TF
Use tabular function specified by CFTFNAME below.

(b) 2 or CF
Use control function specified by CFTFNAME below.

(type = integer / character*2, default = none, units = none)

(8) CFTFNAME
Name of tabular or control function defining the time dependence of the mass addition rate.

(type = character*16, default = none, units = none)

Example

```
RN1_VS 2 !NSTR VOLUME  PHASE CLASS FRACTION MASS     TF/CF
           1  CVolume30 GAS   XE      0.35  9.23     CF CF53
           2  CVolume70 GAS   XE      0.15  9.23     TF TF537
```

3.1.5 Aerosol Modeling Parameters

RN1_ASP – Aerosol Sectional Parameters

Optional.

This record determines the size boundaries and nominal density for the aerosol calculations. If this record is input, and if the aerosol coefficients are read in on the RN1_ACOEF record, the values are checked to see if valid aerosol coefficients have been read in. If not, the code will abort in MELGEN. If the coefficients are calculated, the values on this record will be used to perform the coefficient evaluation.

(1) DMIN
Lower bound aerosol diameter.

(type = real, default = 0.1E-6, units = m)

(2) DMAX
Upper bound aerosol diameter.

(type = real, default = 50.E-6, units = m)

(3) RHONOM
Nominal density of aerosols.

(type = real, default = 1000., units = kg/m^3)

Example

```
!          DMIN            DMAX     RHONOM
RN1_ASP    1.0E-5        0.000005   1000.0
```

RN1_ACOEF – Aerosol Coefficients

Optional.

In general, the aerosol coefficients are calculated by the code. In order to facilitate the use of these coefficients in subsequent calculations, the values can be read in. If the coefficients are calculated as determined by ICOEFF on this record, the coefficient values are written out to the file ACOEFF by MELCOR. The coefficients can be read in with a redirected input file (include) mode.

(1) ICOEFF

Index for the aerosol coefficients calculation:

(a) 1 or CALANDWR
Calculate the coefficients and write them to file ACOEFF.

(b) 0 or CALCUL
Calculate the coefficients, but don't write them to a file (default).

(c) -1 or READ
Read in the coefficients on the table below.

On modern computers, where CPU usage is not a problem, default should be routinely used since it eliminates file handling errors.

(type = integer, default = 0, units = none)

The following fields define the table with aerosol coefficients required if and only if ICOEFF = -1. The length of the table must be equal to 2*NUMSEC*NUMSEC + 4*NUMSEC - 1:

(1) NC

Coefficient index.

(type = integer, default = none, units = none)

(2) CT1P1

Aerosol coefficient for TGAS1 and PGAS1.

(type = real, default = none, units depend on the particular coefficient generated in MELGEN; see the RN Package Reference Manual)

(3) CT2P1

Aerosol dynamics coefficients for TGAS2 and PGAS1.

(type = real, default = none, units depend on the particular coefficient generated in MELGEN; see the RN Package Reference Manual)

(4) CT1P2

Aerosol dynamics coefficients for TGAS1 and PGAS2.

(type = real, default = none, units depend on the particular coefficient generated in MELGEN; see the RN Package Reference Manual)

(5) CT2P2

Aerosol dynamics coefficients for TGAS2 and PGAS2.

(type = real, default = none, units depend on the particular coefficient generated in MELGEN; see the RN Package Reference Manual)

Example

```
RN1_ACOEF READ
!NSTR     CT1P1         CT2P1           CT1P2         CT2P2
 1   0.44007602E+01 0.44659901E+02 0.23426199E+01 0.49471898E+01
 2   0.18199800E+00 0.20193801E+01 0.94409801E-01 0.19795901E+00
 3   0.63367200E+00 0.54719300E+01 0.43126300E+00 0.88422197E+00
```

RN1_PT – Conditions for Aerosol Coefficients

Optional.

This information determines the conditions at which aerosol coefficients are evaluated. The coefficients are evaluated in MELGEN at each of the four combinations of low and high pressures and temperatures specified on this record (i.e., low-T/low-P, low-T/high-P, high-T/low-P, and high-T/high-P). A power law interpolation is then used to calculate values of the aerosol coefficients for intermediate values of pressures and temperatures calculated by the code. If the conditions are outside of the range of these coefficients, the end values are used (i.e., no extrapolation is performed).

(1) PGAS1

Lowest gas pressure for coefficients.

(type = real, default = 1.0E5, units = Pa)

(2) PGAS2

Highest gas pressure for coefficients.

(type = real, default = 2.0E7, units = Pa)

(3) TGAS1

Lowest gas temperature for coefficients.

(type = real, default = 273., units = K)

(4) TGAS2

Highest gas temperature for coefficients.

(type = real, default = 2000., units = K)

Example

```
RN1_PT   9.0E+4  3.0E+5  275.0  398.0 ! Conditions for Aerosol
                                     ! Coefficients
```

RN1_MS00 – Miscellaneous Aerosol Dynamics Constants

Optional.

The miscellaneous coefficients used for the aerosol dynamic processes are input on this record.

(1) CHI

Aerosol dynamic shape factor.

(type = real, default = 1.0, units = none)

(2) GAMMA

Aerosol agglomeration shape factor.

(type = real, default = 1.0, units = none)

(3) FSLIP

Particle slip coefficient.

(type = real, default = 1.257, units = none)

(4) STICK

Particle sticking coefficient.

(type = real, default = 1.0, units = none)

Example

```
RN1_MS00   1.0  1.0   1.257   1.0
```

RN1_MS01 – Miscellaneous Aerosol Dynamics Constants

Optional.

The miscellaneous coefficients used for the aerosol dynamic processes are input on this record.

(1) TURBDS

Turbulence dissipation rate.

(type = real, default = 0.001, units = m^2/s^3)

(2) TKGOP

Ratio of the thermal conductivity of the gas over that for the particle.

(type = real, default = 0.05, units = none)

(3) FTHERM

Thermal accommodation coefficient.

(type = real, default = 2.25, units = none)

(4) DELDIF

Diffusion boundary layer thickness.

(type = real, default = 1.0E-5, units = m)

Example

```
RN1_MS01   0.001   0.05   2.25   1.0E-5
```

RN1_CC – Class/Component Map

Optional.

This record determines the mapping between the material classes, NUMCLS, and the aerosol components, NUMCMP. If NUMCMP = 1 (which is not the default, nor is it recommended), all classes are assigned to component 1. If NUMCMP > 1, the code by default places water class H2O alone in component 2. If NUMCMP > 2, the code by default assigns the volatiles CsI, I_2, Cs, and $CsMoO_4$ to component 3, if those classes are set in the problem. If NUMCMP > 3, there are no default assignments to the additional components. A warning will be issued during initialization if any component has no material classes assigned to it.

The following fields define the table. The length of the table must be equal to NUMCLS. If this record appears, NUMCLS values are required.

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) CLSNAM

The name of material class.

(type = character*16, default = as described, units = none)

(3) CMPNM

Aerosol component number of material class CLSNAM.

(type = integer, default = as described, units = none)

Example

```
RN1_CC ! Number    Class name      Aerosol component number
```

1	XE	2
2	CS	1
3	BA	2
4	I2	2
5	TE	2
6	RU	2
7	MO	3
8	CE	2
9	LA	2
10	UO2	2
11	CD	2
12	AG	2
13	BO2	2
14	H2O	4
15	CON	2
16	CSI	2

RN1_DS – Radionuclide Deposition Surfaces

Optional.

Heat structure surfaces are also deposition surfaces for radionuclides and are defined by HS package input. The RN Package Reference Manual details the default orientations of heat structures as used for deposition purposes in RN. This record is used to override the heat structure information; the orientation can be altered or heat structure surfaces designated as inactive for deposition.

(1) NUMDEP

Number of radionuclide deposition surfaces.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMDEP:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IDS

User heat structure name defined in the HS input.

(type = character*16, default = none, units = none)

(3) ISDE

Desired surface of heat structure.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none)

(4) ITYP

Orientation of heat structure surface for deposition of radionuclides,

(a) 0 or INACTIVE
Not a Deposition Surface.

(b) 1 or CEILING
Ceiling.

(c) 2 or WALL
Vertical Wall.

(d) 3 or FLOOR
Floor.

(type = integer / character*8, default = as defined in HS input, units = none)

Example

```
RN1_DS   4 ! Number Name HS        Surface  Orientation
               1      HS1            LHS     CEILING
               2      HS2            LHS     WALL
               3      HS3            LHS     FLOOR
               4      HS4            LHS     CEILING
```

RN1_TURB - Deposition Modeling Record

Optional

Several options for modeling turbulent deposition in pipes are available in MELCOR. Turbulent deposition is only calculated for those heat structure surfaces specified by the user as calculation of turbulent deposition can impact code performance and is only of importance for high Re number flow in pipes and bends. This record specifies the models that will be used in the calculation of turbulent deposition for those heat structures specified in the RN1_TDS table. A description of the models used in MELCOR for predicting turbulent deposition in pipes and bends is provided in the RN reference manual.

(1) TURBMODEL

Deposition Modeling flag for turbulent component

='OFF' or 0, No turbulent deposition modeling

="VICTORIA' or 1, VICTORIA modeling of deposition in straight pipe sections

='WOODS' or 2, Wood's model for rough pipes

='WOODS_S' or 3, Wood's model for smooth pipes

(type = integer/ character*16, default = 2, units = none)

(2) TRANSMODEL

Deposition Modeling flag for impact deposition in bends and transitions

='OFF' or 0, No deposition modeling in bends

='VICTORIA' or 1, VICTORIA modeling of deposition in bends

='INL' or 2, INL modeling of deposition in bends

='MCF' or 3, McFarland modeling of deposition in bends.

(type = integer/ character*16, default = 2, units = none)

(3) IMODEL

Deposition Modeling flag

=0, Gravitational, thermophoresis, and diffusiophoresis velocities are calculated at the beginning of the calculation

=1, Gravitational, thermophoresis, and diffusiophoresis velocities are recalculated at each time step. Note that if this option is used, it will affect deposition calculated for all deposition mechanism, regardless of whether turbulent deposition is calculated.

(type = integer/ character*16, default = 0, units = none)

Example

```
RN1_TURB  WOODS  2  1
```

RN_TDS Record – RN Turbulent Deposition Surfaces

Optional.

Turbulent deposition modeling can be activated for each surface. Turbulent deposition may be important for high Re flow in a pipe or in pipe bends. If a surface is not defined in this table, it is assumed that turbulent deposition is not calculated. This record characterizes the surfaces associated with this heat structure for the purpose of turbulent deposition. The user supplies characteristic lengths, surface roughness, and the number of bends and angle of bends associated with this structure. The models used for predicting turbulent deposition are defined on the RN1_TURB record. A complete description of the modeling approach is further described in the RN reference manual.

(1)NDEP – Number of deposition surfaces associated with turbulent deposition modeling

The following data are input as a table with length NDEP

(1) NUMTDS – Index for turbulent deposition associated with a particular heat structure surface

(type = integer, default=none, units = none)

(2) HS_ID

The heat structure to apply the bend and/or turbulent deposition model

(type = integer or character*16, default=none, units = none)

(3) ISUR

Surface ('LHS' or 'RHS')to which the deposition modeling is applied

(type = character*3, default=none, units = none)

(4) CHARL

Characteristic length (i.e., pipe diameter)

(type = real, default=none, units = m)

(5) NO_BND

Number of bends associated with the volume

(type = integer, default=none, units = none)

(6) ANGLE

Turning angle of the bends

(type = real, default=none, units = radians)

(7) RAD_BND

Radius of curvature for bend

(type = real, default=none, units = m)

(8) ROUGH

Surface roughness for the turbulent deposition model (not used in VICTORIA model)

(type = real, default=none, units = none)

(9) VelocityFP

The Flow path used to determine flow velocities. This field is optional. If not provided, MELCOR uses the control volume velocity which is calculated from the CV area that is either provided on the CV_ARE record or calculated from the volume divided by the height. If VelocityFP is provided, MELCOR uses the atmosphere velocity for the flow path provided.
(type = integer or character*16, default=none, units = none)

Example

```
RN1_TDS  3
    1      1110   RHS    0.30    1     1.5708     9.00e-2   5.e-5  FP101
    2      1120   RHS    0.063   0     1.5708     9.00e-2   5.e-5
    3    'HS1130'  RHS    0.063   0     1.5708     9.00e-2   5.e-5
```

RN1_SET – Flow-through Areas for Intervolume Transport

Optional.

These records specify the area available for the settling of aerosols from volume to volume without bulk flow. For example, if the floor of a control volume is an arbitrary boundary, specification of a flow-through area will allow aerosols to settle through this arbitrary boundary. This settling is in addition to and independent of transport by intervolume flow.

(1) NUMSET

Number of the flow-through areas for intervolume transport.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMSET:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IVOLF

User control volume name for the *from* volume.

(type = character*16, default = none, units = none)

(3) IVOLT

User control volume name for the *to* volume.

(type = character*16, default = none, units = none)

(4) ELEV

Elevation of the flow-through area.

(type = real, default = none, units = m)

If the water level is above this elevation, no transport occurs.

(5) AREA

Area for the transport.

(type = real, default = none, units = m^2)

Warning: If IVOLF and IVOLT are defined as the same control volume, i.e., the volume settles into itself, physically unrealistic results may be calculated. Currently, when the pool disappears from any such volume the aerosols that were suspended in the liquid are resuspended in the largest section of the gas phase aerosol and allowed to settle where they will. Similarly, vapors that were held in the liquid phase are put into the gas phase. Users are encouraged to always provide a heat structure at the bottom of a control volume onto which vapors and aerosols can deposit. This should result in the most realistic treatment of fission products.

Example

```
RN1_SET 5 ! N  FROM CV  TO CV   ELEVATION    AREA
            1   CV150   CV110      10.0      10.0
            2   CV110   CV120       6.0       4.0
            3   CV900   CV900     -25.0    1000.0
            4   CV111   CV120       6.0       2.0
            5   CV120   CV120       0.0       0.1
```

RN1_AR – Aerosol Resuspension Parameters

Optional.

The size distribution of resuspended aerosols is determined by this input. Because no resuspension models are currently included in MELCOR, this input will not be used.

(1) IDISTP

Distribution parameter.

(a) 1 or UNIFORM

Uniform source with respect to log diameter (i.e., log uniform). No additional values.

(b) 2 or LOGNORMAL

Log-normal distribution with respect to log diameter. Two additional values described below, GEOMM (2) and GSD (3), are needed. The data are input on this data string.

(c) 3 or SECTIONBYSECTION

Section-by-section distribution specified. NUMSEC additional values for FRC below are needed. The data are input from the new line as the table.

(type = integer / character*16, default = 3, units = none)

If IDISTP = 2 or IDISTP = LOGNORMAL then

(2) GEOMM

Source mean diameter (also denoted AMMD). If greater than zero the value is the mass median diameter. If less than zero the absolute value is the geometric mean diameter.

(type = real, default = none, units = m)

(3) GSD

Geometric standard deviation.

(type = real, default = none, units = m)

If IDISTP = 3 or IDISTP = SECTIONBYSECTION then the table with length NUMSEC is input from the new line:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) FRC

Mass fraction of resuspended aerosols in each section. Values are normalized after they are read.

(type = real, default = 1.0, units = none)

Example

```
RN1_AR SECTIONBYSECTION ! NSTR  MASS FRACTION
                            1       2.0
                            2       2.0
                            3       2.0
                            4       2.0
                            5       2.0
```

3.1.6 Condensation/Evaporation Models

RN1_ACD – Aerosol Condensation Index

Optional.

This record determines whether water will condense on all aerosol particles or just on aerosol particles containing water (see Section 2.5.1 of the RN Package Reference Manual). If ICOND = 0 (the default), condensation of water on all aerosol particles will be evaluated. If ICOND = 1, water will condense only on aerosol particles containing water (i.e., CVH fog). For either process, the amount of water that condenses on existing particles can be rate-limited. If this is the case, any excess water that must condense to maintain thermodynamic equilibrium (saturation) conditions is added to the aerosol distribution in the smallest size section. For evaporation, no rate limitation is applied; as much water will evaporate as is necessary to maintain saturation. The total amount of water that condenses or evaporates during a timestep is determined by the CVH package.

(1) ICOND

Index for condensation calculations in aerosol dynamics.

(a) 0 or ALL

Condensation onto all existing aerosols is evaluated (default).

(b) 1 or FOG

Condensation only onto existing water aerosols (CVH fog).

(type = integer, default = 0, units = none)

Example

```
RN1_ACD FOG
```

3.1.7 Decay Heat Distribution

Decay heat generated by radionuclides residing in the control volume atmosphere or on heat structure surfaces in a control volume is distributed among the volume atmosphere, the surfaces of heat structures in that volume, the pool surface (if a pool is present), and the atmospheres and surfaces of other control volumes, as defined by the following input. Splits are the same for all classes.

RN1_DHL – Length for Absorption of Decay Betas

Optional.

Deposition of decay heat in a volume atmosphere results primarily from absorption of beta radiation. The fractions specified by SPVOL1 (record RN1_DHV) or SPSUR2 (record RN1_DHS) are interpreted as the values appropriate for complete absorption. They must be reduced for small volumes or low densities, where the thickness of the atmosphere is insufficient to permit complete absorption of beta rays. The fractions absorbed are based on the range of a typical beta ray (given in sensitivity coefficient array 7002) and the characteristic path for absorption in the control volume, specified on this record. Any reduced deposition is compensated for by proportionate increase in energy distributed to other components specified by the parameters SPVOL2-SPVOL4 or SPSUR3-SPSUR5 entered on the RN1_DHV or RN1_DHS input records or their default values. (The calculation is bypassed if the sum of these other split coefficients is zero.)

(1) NUMLEN

Number of characteristic paths for absorption.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMLEN:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) IVOL

User-defined control volume name.

(type = character*16, default = none, units = none)

(3) CVPATH

Characteristic path for absorption of beta radiation in control volume.

(type = real, default = MIN(VOLUME$^{1/3}$, CVARA$^{1/2}$), units = m)

VOLUME is defined by the volume/altitude table (CV_VAT tabular record) and *CVARA* is the volume flow area (CV_ARE record, or its default).

Example

```
RN1_DHL 2 ! N  Name CVH        Characteristic path
            1  CVolume70              0.01
            2  CVolume30              0.01
```

RN1_DHV – Decay Heat Split for Control Volumes

Optional.

The split of the decay heat from radionuclides in the volume atmosphere is specified on this record. The decay heat may go to any heat structures or other volumes depending on the data. All the decay heat from radionuclides in the water stays in the water phase.

(1) NUMDHV

Number of the decay heat splits for control volumes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMDHV:

(1) NSTR

Data string number.

(type = integer, default = none, units = none)

(2) IVOL

User control volume name.

(type = character*16, default = none, units = none)

(3) SPVOL1

Fraction of the atmosphere decay heat that goes to the control volume atmosphere.

(type = real, default = 0.50, units = none)

(4) SPVOL2

Fraction of the atmosphere decay heat that goes to the surfaces in that volume.

(type = real, default = 0.50, units = none).

Note that surfaces include the heat structures that are exposed to the atmosphere as well as the surface of the pool, and that surface area weighting is used.

(5) SPVOL3

Fraction of the atmosphere decay heat that goes to other volumes.
Optional field.

(type = real, default = 0.0, units = none).

The details of this split are given by the RN1_DHVS record.

(6) SPVOL4

Fraction of the atmosphere decay heat that goes to other heat structure surfaces.
Optional field.

(type = real, default = 0.0, units = none).

The details of this split are given by the RN1_DHVS record.

Example

```
RN1_DHV 2 !N Name   CVH   SPVOL1   SPVOL2   SPVOL3   SPVOL4
           1 CVolume30     0.1      0.8      0.05     0.05
           2 CVolume70     0.1      0.8      0.05     0.05
```

RN1_DHVS – Control Volume Decay Heat Split to Other Components

Optional.

This record is only needed if the RN1_DHV Record specifies that decay heat goes to other control volumes or heat structures not in the control volume where the heat is released, i.e., if SPVOL3 and/or SPVOL4 are non-zero. In this case, this record series specifies which other control volumes or heat structures receive decay heat.

(1) NDHVOL

Number of paths for volume decay heat to other components.

(type = integer, default = 0, units = none)

The following data are input as a table with length NDHVOL:

(1) NSTR

Data string number.

(type = integer, default = none, units = none)

(2) IVOL

User-defined control volume name for the source of decay heat.

(type = character*16, default = none, units = none)

(3) TYPE

Type of the receiving component.
This is either a user-defined control volume or a user-defined heat structure:

(a) 1 or CVH
User-defined control volume.

(b) 2 or HS
User-defined heat structure.

(type = integer / character*3, default = none, units = none).

(4a) ID1

Name of the receiving component.

This is either a user-defined control volume name or a user-defined heat structure name (depends on TYPE above). If the latter, an extra field is required (ID2 below) to specify the left or right surface (thus requiring <u>six</u> fields total for this input line).

(type = character*16, default = none, units = none).

(4b) ID2

If input, ID2 is the pointer to the correct surface of the heat structure.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(5) FRAC

Fraction of the split decay heat that goes to control volume ID1 or heat structure surface ID1/ID2. These values should sum to 1.0 for each volume and/or heat structure subset.

(type = real, default = 0.00, units = none).

Example

```
RN1_DHVS 6 ! N Name CVH   HS/CVH Name      SIDE   FRACTIONS
             1 CVolume30  CVH CVolume35              1.0
             2 CVolume30  HS  HeatSlab3502 LHS       1.0
             3 CVolume70  CVH CVolume67              0.9
             4 CVolume70  HS  HeatSlab67   LHS       0.95
             5 CVolume70  CVH CVolume66              0.1
             6 CVolume70  HS  HeatSlab66   LHS       0.05
```

RN1_DHS – Decay Heat Split for Heat Structures

Optional.

This record determines the split of the decay heat from radionuclides on heat structure surfaces. The decay heat may be split between the heat structure itself, the atmosphere in the associated control volume, other surfaces in that volume, and neighboring volumes and heat structures.

(1) NUMDHS

Number of decay heat splits for heat structures.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMDHS:

(1) NSTR

Data string number.

(type = integer, default = none, units = none)

(2) IDHS1

User-defined heat structure name.

(type = character*16, default = none, units = none)

(3) IDHS2

Pointer to surface of heat structure on which decay occurs.

(a) -1 or LHS

Left side of heat structure.

(b) 1 or RHS

Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(4) SPSUR1

Fraction of the surface decay heat that goes to that surface.

(type = real, default = 0.50, units = none)

(5) SPSUR2

Fraction of the surface decay heat that goes to the atmosphere of the control volume in which the surface resides.

(type = real, default = 0.25, units = none)

(6) SPSUR3

Fraction of the surface decay heat that goes to other surfaces in that volume, including the pool if present, using surface area weighting.

(type = real, default = 0.25, units = none)

(7) SPSUR4

Fraction of the surface decay heat that goes to the atmosphere of other control volumes. Optional field.

(type = real, default = 0.00, units = none).

Details of the split are given by the RN1_DHSS record.

(8) SPSUR5

Fraction of the surface decay heat that goes to heat structure surfaces in other control volumes. Optional field.

(type = real, default = 0.00, units = none).

Details of the split are given by the RN1_DHSS record.

Example

```
RN1_DHS 4 ! N Name HS    SIDE  FRACTIONS
            1 HeatSlab701 RHS   0.6   0.2   0.2
            2 HeatSlab702 LHS   0.6   0.3   0.1
            3 HeatSlab703 LHS   0.6   0.2   0.15   0.03   0.02
            4 HeatSlab704 LHS   0.6   0.2   0.15   0.03   0.02
```

RN1_DHSS – Surface Decay Heat Split to Other Components

Optional.

This record is only needed if the RN1_DHS record specifies that decay heat goes to the atmospheres of other control volumes or to heat structures in other control volumes, i.e., if SPSUR4 and/or SPSUR5 are nonzero. In that case, this record specifies which other volumes or heat structures receive decay heat.

(1) NDHSUR

Number of paths for surface decay heat to other components.

(type = integer, default = 0, units = none)

The following data are input as a table with length NDHSUR:

(1) NSTR

Data string number.

(type = integer, default = none, units = none)

(2) IDHS

User heat structure name for the source of decay heat.

(type = character*16, default = none, units = none)

(3) IDHS2

Pointer to side of heat structure.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(4) TYPE

Type of the receiving component.

This is either a user-defined control volume or a user-defined heat structure:

(a) 1 or CVH
User-defined control volume.

(b) 2 or HS
User-defined heat structure.

(type = integer / character*3, default = none, units = none).

(5a) ID1

Name of the receiving component.

This is either a user-defined control volume name or a user-defined heat structure name (depends on TYPE above). If the latter, an extra field is required (ID2 below) to specify the left or right surface (thus, requiring seven fields total for this input line).

(type = character*16, default = none, units = none).

(5b) ID2

If input, ID2 is the pointer to the correct surface of the heat structure.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(6) FRAC

Fraction of the split decay heat that goes to control volume ID1 or heat structure surface ID1/ID2. These values should sum to 1.0 for each volume and/or heat structure subset.

(type = real, default = 0.00, units = none).

Example

```
RN1_DHSS 6 !N Name HS    SIDE CVH/HS Name   SIDE  FRACTIONS
            1 HeatSlab703 LHS CVH CVolume67          1.0
            2 HeatSlab703 LHS HS  HeatSlab67 LHS     0.9
            3 HeatSlab703 LHS HS  HeatSlab66 LHS     0.1
            4 HeatSlab704 LHS CVH CVolume67          0.8
            5 HeatSlab704 LHS CVH CVolume66          0.2
            6 HeatSlab704 LHS HS  HeatSlab67 LHS     1.0
```

3.1.8 ESF Parameters

RN2_INPUT – Start record for RN2 Package input

Required.

RN2_PLS – Pool Scrubbing

Optional.

Pool scrubbing of aerosols and/or iodine vapor can be treated for all flow paths and flow directions for which the RN SPARC model is enabled (see IBUBF and IBUBT on the FL Package FL_JSW record), and for all gas flows from the CAV model which have RN aerosol and/or iodine vapor scrubbing enabled (see input of IBUBX on record CAV_U). This input allows the user to vary the parameters used in each such application of the model.

(1) NUMPLS

Number of the pool scrubbing flow paths.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMPLS:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) FPCAV

Regular flow paths and cavity water pools can be handled as discussed below:

(a) -1 or FROMFP
The pool scrubbing parameters apply in the *from* volume.

(b) 1 or TOFP
The pool scrubbing parameters apply in the *to* volume.

(c) 3 or CAVPOOL
A cavity water pool.

(type = integer / character*7, default = none, units = none)

(3) NAME

Pool scrubbing user flow path name or cavity name.

(Depends on the value of FPCAV above.)

(type = character*16, default = none, units = none)

(4) AVENT

Flow area of the vent (or individual holes in multihole vents).

(type = real, default = *FLARA* (FL Package) for regular flow paths and a CORCON calculated value for cavity water pools, units = m^2)

(5) MVENT

Type of vent (1=multi-hole sparger/quencher, 2=downcomer, 3=horizontal vent).

(type = integer, default = 3, units = none)

(6) NVENT

Number of holes in the vent (usually 1 except for vents of type MVENT=1).

(type = integer, default = 1, units = none)

(7) NTYPE

Number of identical vents in use (normally 1).

(type = integer, default = 1, units = none)

Example

```
RN2_PLS 3 ! N FP/CAV  NAME    AVENT   MVENT NVENT NTYPE
            1 ToFP    FL199   0.64E-4   1    500    1
            2 ToFP    FL320   0.64E-4   1    500    1
            3 CavPool CAV7   -1.0       3     1     1
```

RN2_FLT – Filter Parameters

Optional.

This record gives the location of the filter and the performance characteristics for the filter. Three fields are required; the fourth through seventh fields are optional.

(1) FLTNAM

User-defined filter name.

(type = character*16, default = none, units = none)

(2) IFLTFP

Filter user flow path name. The filter is located in this flow path for aerosol and fission product vapor removal.

(type = character*16, default = none, units = none)

(3) CTYPE

Type of filter. Removal of either aerosols or fission product vapors is given by this parameter.

(a) AEROSOL
Aerosols are removed

(b) FPVAPOR
Removes fission product vapors

(type = character*7, default = none, units = none)

(4) IKEY

Key if control function specifies the global decontamination factor for the filter or value

(a) -1 or CF
Use CF.

(b) 0 or VALUE
Use value.

(type = integer / character*5, default = VALUE, units = none)

optionally followed by fifth field

(5) DFG

Global decontamination factor for the filter.

(type = real, default = 1.0, units = none)

or

(5) NameCF

User name of a control function that specifies the global decontamination factor for the filter.

Note that DFG ≡ (mass entering filter) / (mass not removed by filter) must be ≥ 1.0. If a control function is specified, its value must be appropriate.

This decontamination factor is used for all classes but water, except as modified below by the RN2_FCL record. The value for the water class is 1.0, regardless of the value of DFG, unless another value is specified for the water class on a RN2_FCL record.

(type = character*16, default = none, units = none)

(6) XMASG

Limit on total mass removed by the filter. If the value is negative, there is no total mass limit.

(type = real, default = – 1.0, units = kg)

(7) ICVTYP

Control volume type associated with the filter for editing and accounting purposes. Normally, this entry should be the same as the *ICVTYP* entry on the CV_ID record associated with the control volume from which the filter removes fission products (see CVH Package Users' Guide).

(type = integer, default = TYPE99 or FILTERS, units = none)

Examples

```
RN2_FLT 'Filter1' 'FL398' 'AEROSOL' VALUE 1000.0 -1.0 CVType99
RN2_FLT 'Filter2' 'FL398' 'FPVAPOR' VALUE 1000.0 -1.0 CVType99
RN2_FLT 'Filter3' 'FL398' 'FPVAPOR' CF    CF130  -0.1 CVType99
```

The following set of records may be input for each filter. If any records are input, they must be placed after the corresponding RN2_FLT record.

RN2_FCL – Filter Class Parameters

Optional.

This record modifies the global parameters specified on the previous record for specific classes. The data are input in the form of table. Two fields are required. The third through fifth fields are optional in each table line.

(1) NUMCLS

Number of classes to be modified.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMCLS:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLSS

Class name.

(type = character*24, default = none, units = none)

(3) IKEY

Key if control function specifies the global decontamination factor for the filter or value

(a) -1 or CF
Use CF.

(b) 0 or VALUE
Use value.

(type = integer / character*5, default = VALUE, units = none)

optionally followed by fourth field

(4) DFC

Class decontamination factor. (Note: DFC must be ≥1.0).

(type = real, default = DFG on RN2_FLT record, units = none)

or

(4) NameCF

User name of a control function that specifies the class decontamination factor. The value of the control function must be ≥ 1.0.

(type = character*16, default = none, units = none)

(5) XMASC

Limit on class mass removed by the filter (see description for XMASG on the previous record).
Optional field.

(type = real, default = -1.0, units = kg)

Example

```
RN2_FCL 1  ! N ClassName    Key   DFC/CF   XMASC
             1    XE       VALUE    1000. -0.1
             2    BA         CF    CF130   -0.1
```

RN2_FDA – Decontamination Factor as a Function of Aerosol Size

Optional, permitted only for the AEROSOL filter.

This record gives the performance of the filter as a function of aerosol size. The value is applied to all classes and takes precedence over input on the RN2_FLT record and on RN2_FCL tabular record. The value for the water class in section ISEC is 1.0, regardless of the value of *DFGS* unless another value is specified for the water class in this section on a RN2_FDC record.

(1) NUMROW

Number of user input rows.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMROW. All four fields are required:

(1) N

The number of user input row.

(type = integer, default = none, units = none)

(2) ISEC

Aerosol section (size bin) specifying aerosol size.

(type = integer, default = none, units = none)

(3) IKEY

Key: If control function specifies the sectional decontamination factor for the filter for section ISEC or value

(a) -1 or CF
Use CF.

(b) 0 or VALUE
Use value.

(type = integer / character*5, default = none, units = none)

optionally followed by

(4) DFGS

Decontamination factor for the filter for section ISEC.

(type = real, default = none, units = none)

or

(4) IDFGS

User name of a control function that specifies the sectional decontamination factor for the filter for section ISEC.

(type = character*16, default = none, units = none)

Example

```
RN2_FDA   2  ! N Section Key     DecFactor/NameCF
               1      3       VALUE   2.35
               2      4       CF      CF130
```

RN2_FDC - Decontamination Factor as a Function of RN Class and Aerosol Size

Optional, permitted only for the AEROSOL filter.

This record gives the performance of the filter as a function of aerosol size. The value takes precedence over input on the RN2_FLT record, RN2_FCL record, and RN2_FDA record.

(1) NUMROW

Number of user input rows.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMROW. All five fields are required:

(1) N

The number of user input row.

(type = integer, default = none, units = none)

(2) ISEC

Aerosol section (size bin) specifying aerosol size.

(type = integer, default = none, units = none)

(3) ICLS

RN class name.

(type = character*16, default = none, units = none)

(4) IKEY

Key: if control function specifies the decontamination factor for the filter for class ICLS in section ISEC or value

(a) -1 = CF
Use CF.

(b) 0 = VALUE
Use value.

(type = integer / character*5, default = none, units = none)

optionally followed by

(5) DFCS

Decontamination factor for the filter for class ICLS in section ISEC.

(type = real, default = none, units = none)

or

(5) IDFCS

User name of a control function that specifies the sectional decontamination factor for the filter for class ICLS in section ISEC.

(type = character*16, default = none, units = none)

Example

```
RN2_FDC   2  ! N Section RN class Key    DecFactor/NameCF
           1      5       CS       VALUE  3.56
           2      5       H2O      CF     CF130
```

RN2_FIM – Initialization of Filter RN Mass

Optional.

This record allows initialization of masses for RN classes.

(1) NUMCLS

Number of classes to be modified.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMCLS:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLSS

RN class name for which an initial mass is to be placed on filter FLTNAM (record RN2_FLT).

(type = character*16, default = none, units = none)

(3) XMFLTI

Initial (total) mass of ICLSS material.

(type = real, default = 0.0, units = kg)

(4) XMFRAC

Fraction of XMFLTI that is radioactive.

(type = real, default = 1.0, units = none)

Example

```
RN2_FIM   1  !   N   ICLSS XMFLTI XMFRAC
                 1     CS   3.56    1.0
```

Optional records RN2_FAC through RN2_FCD below activate models that represent the specific characteristics of specialized filters. These models (developed for application of MELCOR to non-LWR plants) are not currently described in the RN Package Reference Manual. A value JSEG (that identifies the flow path segment within which filter FLTNAM resides) on the RN2_FAC record must also be provided in FL Package input, on record FL_SEG for flow path name IFLTFP (such segment should exist in the flow path).

RN2_FAC – Performance Characteristics of Aerosol Filters

Optional, may be input for filters of AEROSOL type (CTYPE on RN2_FLT record).

(1) XMNSZA

Minimum aerosol size (mean diameter) for which decontamination factors defined by Records RN2_FLT and RN2_FCL apply. Aerosols with smaller diameters will not be affected by the filter.

(type = real, default = 0.0, units = m)

(2) FAILDP

Pressure drop (segment pressure loss) at which the filter fails. On failure, the filter decontamination factor is reset to 1.0. A fraction FAILRL of the RN mass deposited on the given filter is released, and the value for SLAM input on record FL_SEG is reset to 0.

(type = real, default = -1.0, units = Pa)

(3) FAILRL

Fraction of RN mass (applied to all classes, and to both radioactive and nonradioactive fractions) that will be released from the given filter on failure. The released mass will be evenly distributed among the sections of the aerosol distribution above diameter *XMNSZA*.

(type = real, default = 1.0, units = none)

(4) DHRELS

Fraction of the total decay heat generated by RN material (of all classes) on the given filter that contributes to heating the gases passing through the filter. The remaining fraction (1-DHRELS) is assumed to be lost to the environment.

(type = real, default = 1.0, units = none)

(5) JSEG

Flow path segment number within which the given filter resides.

(type = integer, default = 0, units = none)

Example

```
!         XMNSZA   FAILDP   FAILRL   DHRELS   JSEG
RN2_FAC    0.0       -1.    1.       1.      2
```

Or

```
RN2_FAC    0.0      FailP   RelFP    1.      2
```

Where FailP is the logical control function for the FAILDP input, where as RelFP is the real control function name for FAILRL input.

RN2_FVC – Performance Characteristics of Charcoal (Vapor) Filters

Optional, may be input for filters of FPVAPOR type (CTYPE on RN2_FLT record)

From one to four values may be entered; unspecified parameters retain their default values.

(1) AGE

Service life of charcoal filter.

(type = real, default = 0.0, units = months)

(2) XMBEDI

Initial mass of charcoal in filter.

(type = real, default = 827.35, units = kg)

(3) DHRELS

Fraction of the total decay heat generated by RN material (of all classes) on the given charcoal filter which goes to heating the charcoal and the gases passing through the filter. The remaining fraction (1-DHRELS) is assumed to be lost to the environment. For most applications, this value should be the same as the value of DHRELS used for the aerosol filter.

(type = real, default = 1.0, units = none)

(4) SPECHT

Specific heat of charcoal

(type = real, default = 711.78, units = J/kg K)

Example

```
!         AGE   XMBEDI   DHRELS  SPECHT
RN2_FVC   0.0   827.35     1.   711.78
```

RN2_FTD – Thermal Desorption Data for Charcoal Filters

Optional, may be input for filters of FPVAPOR type (CTYPE on RN2_FLT record).

If absent, thermal desorption is not modeled. One or both values may be entered.

(1) ITHCOF

Thermal desorption coefficient index

(a) 0 or NO

No thermal desorption. No additional fields are needed.

(b) -1 or FISH6

Use correlation for thermal desorption rate from the FISH6 code.

(c) -2 or ROOM

Use correlation for thermal desorption rate from the ROOM code.

(d) 1 or TF

Use tabular function *TFNAME* to specify desorption rate (fraction released / s) as a function of temperature (K).

(type = integer / character*5, default = 0, units = none)

optionally followed by

(2) THAGE

Age coefficient index.

(a) = 0.

Use expression consistent with correlation specified by *ITHCOF*.

(b) > 0.

Use constant age coefficient THAGE.

(type = real, default = 0.0, units = none

or

(2) TFNAME

Tabular function name.

(type = character*16, default = none, units = none

Examples

```
RN2_FTD NO            ! No additional fields are required
RN2_FTD FISH6         ! THAGE == 0 (default) is used
RN2_FTD FISH6  10.0   ! THAGE /= 0 is used
RN2_FTD ROOM          ! THAGE == 0 (default) is used
RN2_FTD ROOM   10.0   ! THAGE /= 0 is used
RN2_FTD TF     TF1    ! tabular function
```

RN2_FRD – Radiolytic Desorption Data for Charcoal Filters

Optional, may be input for filters of AEROSOL type (CTYPE on RN2_FLT record).

If absent, radiolytic desorption is not modeled. One to three values may be entered.

(1) IRDCOF

Radiolytic desorption coefficient index.

(a) 0 or NO
Radiolytic desorption is not modeled. No additional fields are needed.

(b) -2 or ROOM
Use correlation for radiolytic desorption rate from the ROOM code.

(c) 1 or TF
Use tabular function *TFNAME* to specify desorption rate (fraction released / s) as a function of exposure.

(type = integer / character*6, default = none, units = none)

optionally followed by

(2) RADAGE

Age coefficient index. Optional field.

(a) = 0.
Use the expression for the age coefficient in the ROOM code.

(b) > 0.
Use constant age coefficient RADAGE.

(type = real, default = 0, units = none)

or

(2) TFNAME

Tabular function name. Required if IRDCOF = TF or 1.

(type = character*16, default = none, units = none

(3) EXTIMI

Initial radiation exposure time. Optional field.

(type = real, default = 0.0, units = s)

Examples

```
RN2_FRD NO                ! no additional fields are required
RN2_FRD Room              ! RADAGE, EXTIMI by default
RN2_FRD Room  1.0         ! EXTIMI by default
```

```
RN2_FRD Room  1.0     2.0 ! no defaults
RN2_FRD TF TF1            ! EXTIMI by default
RN2_FRD TF TF2  1.0       ! EXTIMI > 0
```

RN2_FCD – Charcoal Combustion Data for Charcoal Filters

Optional, may be input for filters of AEROSOL type (*CTYPE* on RN2_FLT record).

If absent or BURNRT = 0.0, charcoal combustion is not modeled. All fields are optional. If the record is input and BURNRT = 0.0, then all other fields are not used.

(1) BURNRT

Burn rate expression index.

(a) = 0.

Charcoal burning is not modeled.

(b) =-1.

Use expression for burn rate described in the RN Reference Manual (from coal conversion technology).

(c) > 0.

Constant burn rate of BURNRT (kg/m^2-s).

(type = real, default = 0.0, units = none or kg/m^2-s)

(2) TMPIGN

Charcoal ignition temperature.

(type = real, default = 573.0, units = K)

(3) APPRHO

Apparent charcoal particle density.

(type = real, default = 520.6, units = kg/m^3)

(4) AVPDIA

Average charcoal particle diameter.

(type = real, default = 0.0016, units = m)

(5) XMFREM

Mass of charcoal remaining after burn is complete. Burn then consumes *XMBEDI – XMFREM*.

(type = real, default = 0.0, units = kg)

(6) PROCON

Proportionality coefficient between fraction of initial charcoal mass consumed by burning and fraction of iodine mass on filter (at time of ignition) that is released (i.e., ratio of charcoal combustion rate to iodine release rate) PROCON > 1.0.

(type = real, default = 1.0, units = none)

Example

```
RN2_FCD 0.
```

RN2_CNV – Options

Optional.

(1) ICONV

Convection option switch.

(a) 0 or CALCCONV

Flow path convection of radionuclides will be calculated.

(b) 1 or NOCONV

Flow path convection will not be calculated.

(type = integer / character*8, default=0, units = none)

Example

```
RN2_CNV  0  ! CALCULATED CONVECTION
```

3.1.9 Radionuclide Chemistry

RN1_RCT – Class Reaction Information

Optional.

These records specify the chemical reactions that will be considered in MELCOR for vapors in the gas phase. The stoichiometric reaction is specified, and forward or reverse reactions may occur depending on the values of the reaction mass transfer data. For evaluation of the reacted mass, the total rate based on mass transfer plus the specified reaction rate is used. In addition, a deposition velocity may be input which is the total rate including mass transfer. The energy associated with the reaction may also be specified. These rates may be input as control functions or tabular functions depending on the complexity desired.

The first *from* class is the fission product vapor in the atmosphere that undergoes the reaction. All other from classes as well as the *to* classes are assumed to be on the surface. The reactions are limited by the availability of the appropriate classes. For example, if the reaction is a forward reaction, and, for example, the second *from* class is not on the surface, the reaction will not take place. Remember, all states must be vapor states as only vapors undergo reactions – aerosol states do not react but only act as surfaces where the reactions can occur.

(1) RCTNAM

Chemical reaction name.

(type = character*16, default = none, units = none)

Example

```
RN1_RCT  ChemReac1  ! Chemical reaction name
```

RN1_RCT00 – *From* Class Specification

Required if RN1_RCT record appears.

(1) NRCLF

Number of *from* classes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NRCLF:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLASF

The *from* reacting class name.

(type = character*16, default = none, units = none)

(3) FRACF

The mole fraction for this class based on a stoichiometric reaction.

(type = real, default = 0.0, units = none)

As many data strings as necessary may be specified.

Example

```
RN1_RCT00 2 ! N    FROM CLASS NAME  MOLE FRACTION
              1          'CSI'           0.2
              2          'BA'            0.25
```

RN1_RCT01 – *To* Class Specification

Required if RN1_RCT record appears.

(1) NRCLT

Number of *to* classes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NRCLT:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLAST

The *to* reacting class name.

(type = character*16, default = none, units = none)

(3) FRACT

The mole fraction for this class based on a stoichiometric reaction.

(type = real, default = 0.0, units = none)

As many data strings as necessary may be specified.

Example

```
RN1_RCT01 2 ! N    TO CLASS NAME  MOLE FRACTION
              1         'H2O'          0.3
              2         'AG'           0.3
```

RN1_RCT02 – Translation Array

Required if RN1_RCT record appears.

The mass fraction of the *from* classes forming each of the *to* classes is specified by this record. This information is needed to properly allocate the radioactive masses. The data are input in the *form* of table. The number of required mass fraction is the number of *from* classes times the number of *to* classes. The number of rows must be equal to the number of *from* classes. First row starts from the line next to header line. Each row must contain (1 + number of to classes) data fields. They are

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) FRACFT

The mass fraction of the *from* class forming each of the *to* classes. The order is the fraction of this *from* class to all of the *to* classes.

(type = real, default = 0.0, units = none)

Example

For example, given the reaction A + B $\rightarrow$ C + D, the order of the input would be

```
RN1_RCT02  !  to classes: C and D
         1    FRACFT_AC     FRACFT_AD    ! from class A
         2    FRACFT_BC     FRACFT_BD    ! from class B
```

For example, consider the reaction

$$H_2O + CO \rightarrow CO_2 + H_2.$$

The mass fraction input would be 0.89 (weight of O divided by weight of H_2O), 0.11, 1.0, or 0. The number of required mass fractions is the number of *from* classes times the number of *to* classes.

```
RN1_RCT02  ! to classes: CO2 and H2
      1    0.89       0.11       * from class H2O
      2    1.0        0.0        * from class CO
```

RN1_RCT03 – Control Volumes in which the reaction may occur

Optional.

This record defines the control volumes in which the reaction may occur. More than one CV can be given on this record.

(1) NPVOL

Number of control volumes in which the reaction may occur.

(type = integer, default = 0, units = none)

The following data are input as a table with length NPVOL:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) IVOL

The control volume name.

(type = character*16, default = none, units = none)

Example

```
RN1_RCT03  7 ! N    CV NAME
               1    CVolume40
               2    CVolume41
               3    CVolume42
               4    CVolume43
               5    CVolume44
               6    CVolume45
               7    CVolume46
```

RN1_RCT04 – States where the reaction can occur

Optional.

This record defines the states within the control volumes where the reaction can occur. As many states as necessary may be specified.

(1) NPSTT

Number of states within the control volumes where the reaction can occur.

(type = integer, default = 0, units = none)

The following data are input as a table with length NPSTT:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ISTAT

The states within the control volumes where the reactions can occur

(a) 1 or HS

Vapors on the HS NAMEHS on side SIDE (input below; thus two additional fields (2a), (2b) are required).

(b) 2 or AG

Aerosols in the vapor phase.

(c) 3 or LP

Liquid pool.

(type = integer / character*2, default = none, units = none)

(2a) NAMEHS

User heat structure name. Required if ISTAT = 1 or HS.

(type = character*16, default = none, units = none).

(2b) SIDE

Specification of the left or right side of heat structure NAMEHS. Required if ISTAT = 1 or HS.

(a) -1 or LHS

Left side of heat structure.

(b) 1 or RHS

Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(3) IRSTTW

Flag to determine if the reaction will occur if there is water present on or in the reaction medium ISTAT (heat structure, atmosphere, or pool).

(c) 0 or STOP
Reaction will stop if water is present.

(d) 1 or CONTINUE
Reaction will continue if water is present.

(type = integer / character*8, default = 0, units = none)

Example

```
RN1_RCT04 5 ! N  STATE  [NAMEHS    SIDE]   FLAG
              1  HS      HeatSlab45 LHS    CONTINUE
              2  HS      HeatSlab46 LHS    CONTINUE
              3  AG                        CONTINUE
              4  LP                        CONTINUE
              5  HS      HeatSlab41 LHS    CONTINUE
```

RN1_RCT10 – Forward Reaction Mass Transfer Data

Optional.

(1) CFMF

Location of the forward reaction mass transfer data

(a) 1 or TF
The data are given as a function of time by tabular function name *NAMECFTF*.

(b) 0 or NOREACT
No forward reaction.

(c) -1 or CF
The data are given by control function name *NAMECFTF*.

(type = integer / character*8, default = 0, units = none)

(2) NAMECFTF

Tabular function name or control function name. Input depends on CFMF value. Required if CFMF equals TF or 1, or CF or –1 only.

(type = character*16, default = none, units = none).

Note: If the value of the function is positive, the value is a rate constant which does not include mass transfer effects and the units of the function must be fraction per second. If the value of the function is negative or zero, the absolute value is a deposition velocity in m/s which includes mass transfer effects and is a net transfer rate. See the RN Package Reference Manual for further details.

Example

```
RN1_RCT10 TF TF916  ! forward reaction mass transfer data
```

RN1_RCT11 – Backward Reaction Mass Transfer Data

Optional.

(1) CFMB

Location of the backward reaction mass transfer data.

(a) 1 or TF
The data are given as a function of time by tabular function name NAMECFTF.

(b) 0 or NOREACT
No backward reaction.

(c) -1 or CF
The data are given by control function name NAMECFTF.

(type = integer / character*8, default = 0, units = none)

(2) NAMECFTF

Tabular function name or control function name. Input depends on CFMB value. Required if CFMB equals TF or 1, or CF or –1 only.

(type = character*16, default = none, units = none).

Note: The value of the function is the reverse reaction constant in units of kg/s. If the forward reaction data is a deposition velocity, this value is not used. See the RN Package Reference Manual for further details.

Example

```
RN1_RCT11 TF TF916  ! Backward reaction mass transfer data
```

RN1_RCT20 – Forward Reaction Energy Transfer Data

Optional.

(1) CFEF

Location of the forward reaction energy transfer data.

(a) 1 or TF
The data are given as a function of time by tabular function name NAMECFTF.

(b) 0 or NOREACT
No energy transfer for the forward reaction.

(c) -1 or CF
The data are given by control function name NAMECFTF.

(type = integer / character*7, default = 0, units = none)

(2) NAMECFTF
Tabular function name or control function name. Input depends on *CFEF* value. Required if *CFEF* equals TF or 1, or CF or –1 only.

(type = character*16, default = none, units = none).

Note: The units of the function are J per kg of mass transferred of the first from class. The energy is added to the atmosphere in the case of aerosols, to the pool for reactions in the pool, and to the heat structure for the heat structure case.

Example

```
RN1_RCT20 TF TF918  ! Forward reaction energy transfer data
```

RN1_RCT21 – Backward Reaction Energy Transfer Data

Optional.

(1) CFTO
Location of the backward reaction energy transfer data.

(a) 1 or TF
The data are given as a function of time by tabular function name NAMECFTF.

(b) 0 or NOREACT
No energy transfer for the backward reaction.

(c) -1 or CF
The data are given by control function name NAMECFTF.

(type = integer / character*7, default = 0, units = none)

(2) NAMECFTF
Tabular function name or control function name. Input depends on CFTO value. Required if CFTO equals TF or CF only.

(type = character*16, default = none, units = none).

Note: See the above record for discussion of the units of the function.

Example

```
RN1_RCT20 TF TF914  ! Backward reaction energy transfer data
```

RN1_TRN – Class Transfer Information

Optional.

These records specify the class transfers that will be considered in MELCOR to simulate rapid chemical reactions. The stoichiometric reaction is specified, and only forward transfers may occur. The energy associated with the transfer may also be specified. These rates may be input as control functions or tabular functions depending on the complexity desired. These records can also be used to change states of masses, i.e., to change from deposited aerosols, which do not undergo chemical reactions, to deposited vapors, which can react.

(1) CLSTRNAM

Class transfer name.

(type = character*16, default = none, units = none)

Example

```
RN1_TRN ClassTransfer1  ! Transfer name
```

RN1_TRN00 – *From* Class Specification

Required if RN1_TRN record appears.

(1) NTRFP

Number of *from* classes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NTRFP:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLASF

The *from* transfer classes name.

(type = character*16, default = none, units = none)

(3) FRACF

The mole fraction for this class based on a stoichiometric reaction.

(type = real, default = 0.0, units = none)

As many data strings as desired may be specified.

Example

```
RN1_TRN00 2 ! N    FROM CLASS NAME  MOLE FRACTION
```

```
          1          'TE'          0.4
          2          'BA'          0.7
```

RN1_TRN01 – *To* Class Specification

Required if RN1_TRN record appears

(1) NTRTP

Number of *to* classes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NTRTP:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICLAST

The *to* transfer classes name.

(type = character*16, default = none, units = none)

(3) FRACF

The mole fraction for this class based on a stoichiometric reaction.

(type = real, default = 0.0, units = none)

As many data strings as desired may be specified.

Example

```
RN1_TRN01  2 ! N    TO CLASS NAME  MOLE FRACTION
               1          'AG'          0.3
               2          'CD'          0.4
```

RN1_TRN02 – Translation Array

Required if RN1_TRN record appears.

The mass fraction of the *from* classes forming each of the *to* classes is specified by this record. This information is needed to properly allocate the radioactive masses. The data are input in the form of table. The number of required mass fraction is the number of *from* classes times the number of *to* classes. The number of rows must be equal to the number of *from* classes. The first row starts from the line next to header line. Each row must contain (1 + number of *to* classes) data fields. They are

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) FRACFT

The mass fraction of the *from* class forming each of the *to* classes. The order is the fraction of this *from* class to all of the *to* classes.

(type = real, default = 0.0, units = none)

Example

For the given the reaction A + B → C + D, the order of the input would be

```
RN1_TRN02 ! ToCLASS 1     ToCLASS 2   to classes: C and D
      1    FRACFT_AC     FRACFT_AD  ! from class A
      2    FRACFT_BC     FRACFT_BD  ! from class B
```

For example, consider the reaction

$$H2O + CO \rightarrow CO2 + H^2.$$

The mass fraction input would be 0.89 (weight of O divided by weight of H_2O), 0.11, 1.0, or 0. The number of required mass fractions is the number of *from* classes times the number of *to* classes.

```
RN1_TRN02 ! ToCLASS 1  ToCLASS 2 ! to classes: CO2 and H2
      1    0.89       0.11       ! from class H2O
      2    1.0        0.0        ! from class CO
```

RN1_TRN03 – Control Volumes in which the transfers may occur

Optional.

This record defines the control volumes in which the transfers may occur. More than one CV can be given on this record:

(1) NTRVP

Number of control volumes in which the reaction may occur.

(type = integer, default = 0, units = none)

The following data are input as a table with length NTRVP:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) IVOL

The control volume name.

(type = character*16, default = none, units = none)

Example

```
RN1_TRN03  1 ! N    CV NAME
               1    Cvolume40
```

RN1_TRN04 – States where the transfers can occur

Optional.

This record defines the states within the control volumes where the transfers can occur. As many states as necessary may be specified:

(1) NTSTP

Number of states within the control volumes where the transfers can occur.

(type = integer, default = 0, units = none)

The following data are input as a table with length NTSTP:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ISTATF

The *from* state within the control volume where the transfers can occur.

(a) -1 or ALL
All states, the *from* states and *to* states are the same.

(b) 1 or HSA
Aerosols on HS NAMEHS on side SIDE (input below; thus, two additional fields (2a), (2b) are required).

(a) 2 or HSV
Vapors on HS NAMEHS on side SIDE (input below; thus, two additional fields (2a), (2b) are required).

(b) 3 or AG
Aerosols in the vapor phase.

(c) 4 or AL
Aerosols in the liquid phase.

(d) 5 or VG
Vapors in the vapor phase.

(e) 6 or VL
Vapors in the liquid phase.

(type = integer / character*3, default = none, units = none)

(2a) NAMEHS

User heat structure name. Required if ISTATF equals HSA or 1 or HSV or 2.

(type = character*16, default = none, units = none).

(2b) SIDE

Specification of the left or right side of heat structure NAMEHS. Required if ISTATF equals HSA or 1 or HSV or 2.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(3) ISTATT

The *to* state within the control volume where the transfer can occur:

(a) -1 or ALL
All states. The *from* states and *to* states are the same.

(b) 1 or HSA
Aerosols on HS NAMEHS on side SIDE (input below; thus, two additional fields (3a), (3b) are required).

(c) 2 or HSV
Vapors on HS NAMEHS on side SIDE (input below, thus two additional fields (3a), (3b) are required).

(d) 3 or AG
Aerosols in the vapor phase.

(e) 4 or AL
Aerosols in the liquid phase.

(f) 5 or VG
Vapors in the vapor phase.

(g) 6 or VL
Vapors in the liquid phase.

(type = integer / character*3, default = none, units = none)

(3a) NAMEHS

User heat structure name. Required if ISTATF equals HSA or 1 or HSV or 2.

(type = character*16, default = none, units = none).

(3b) SIDE

Specification of the left or right side of heat structure NAMEHS. Required if ISTATF equals HSA or 1 or HSV or 2.

(a) -1 or LHS
Left side of heat structure.

(b) 1 or RHS
Right side of heat structure.

(type = integer / character*3, default = none, units = none).

(4) ITSTTW

Flag to determine if the transfer will occur if there is water present for the above states.

(a) 0 or CONTINUE
Transfer will continue if water is present.

(b) 1 or STOPFROM
Transfer will stop if water is present for the *from* state.

(c) 2 or STOPTO
Transfer will stop in water is present for the *to* state.

(d) 3 or STOPEITHER
Transfer will stop if water is present for the either state.

(e) 4 or STOPBOTH
Transfer will stop if water is present for the both states.

(type = integer / character*10, default = 0, units = none)

As many sets as desired may be specified.

Example

```
RN1_TRN04  1 ! N   FROM STATE  [HS]  TO STATE  [HS] FLAG
               1  VG                AG            CONTINUE
```

RN1_TRN10 – Mass Transfer Data

Optional.

(1) CFM

Location of the mass transfer data.

(a) 1 or TF
The data are given as a function of time by tabular function name NAMECFTF.

(b) 0 or NOTRANSFER
No transfer, (no mass or energy transfer).

(c) -1 or CF

The data are given by control function name NAMECFTF.

(type = integer / character*8, default = 0, units = none)

(2) NAMECFTF

Tabular function name or control function name. Input depends on *CFM* value. Required if *CFM* equals TF or 1, or CF or –1 only.

(type = character*16, default = none, units = none).

Note: The units of the function are kg/sec of the first *from* class. If the function value is negative, all the mass is transferred. No reverse transfer is permitted.

Example

```
RN1_TRN10 TF TF915  ! Mass transfer data
```

RN1_TRN11 – Energy Transfer Data

Optional.

(1) CFE

Location of the energy transfer data.

(a) 1 or TF
The data are given as a function of time by tabular function name NAMECFTF.

(b) 0 or NOTRANSFER
No transfer.

(c) -1 or CF
The data are given by control function name NAMECFTF.

(type = integer / character*8, default = 0, units = none)

(2) NAMECFTF

Tabular function name or control function name. Input depends on *CFE* value. Required if *CFE* equals TF or 1, or CF or –1 only.

(type = character*16, default = none, units = none).

Note: The units of the function are J per kg of mass transferred <u>of the first *from* class</u>. The energy is added to the atmosphere in the case of aerosols, to the pool for reactions in the pool, and to the heat structure for the heat structure case.

Example

```
RN1_TRN11 TF TF919  ! Energy transfer data
```

3.1.10 Chemisorption

RN1_CAF – Chemisorption Activation Flag

Optional.

(1) ICAON

Chemisorption activation flag, 1 or ON, 0 or OFF.

(type = integer / character*3, default = 1, units = none)

Example

```
RN1_CAF OFF ! Chemisorption activation flag
```

RN1_CAON – Chemisorption Class Activation Flags

Optional.

This is the CA (Chemisorption Activation) class card, used to set CA classes on or off and to set up new classes.

(1) NNCA

Number of CA classes.

(type = integer, default = 0, units = none)

The following data are input as a table with length NNCA:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) ICACL

CA class number.

(type = integer, default = none, units = none)

(3) ICACON

CA class flag, 0=ON, 1=OFF.

(type = integer / character*3, default = see below, units = none)

(4) ICAST

Surface material type for CA class.

(a) 1 or STAINLESSSTEEL

Stainless steel.

(b) 2 or INCONEL

Inconel (currently not in material database)

(c) 3 or ZIRCALOY
Zircaloy.

(type = integer / character*14, default = see below, units = none)

Note: Chemisorption currently can only be done on materials in the MELCOR database and does not function for user-defined materials. Currently, Inconel is not in the database. Also, note that the CsOH class must be ON if the CsI class is ON, as iodine released in the CsI chemisorption reaction is released via the CsOH vapor deposition class. The first four classes are default ON, and the next two (tellurium) are OFF because current data suggests that tellurium does not chemisorb in the presence of an oxide film on the surface.

Example

```
RN1_CAON 1 ! N  CA CLASS NUMBER  FLAG  MATERIAL TYPE
             1         2         OFF   STAINLESSSTEEL
```

RN1_CACM – Chemisorption Class to Radionuclide Vapor Mapping

Optional.

(1) NNCAM
Length of chemisorption class to radionuclide vapor mapping table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NNCAM:

(1) N
Data string number.

(type = integer, default = none, units = none)

(2) ICA
CA class number.

(type = integer, default = none, units = none)

(3) ICL
Radionuclide vapor class name.

(type = character*16, default = none, units = none)

Example

```
RN1_CACM 1 ! N  CA CLASS NUMBER  RN CLASS NAME
             1         1              XE
```

3.1.11 Iodine Pool Model

RN1_IOD01 – Iodine model activation flags

Optional.

If this record appears then first field is required, the others are optional. Fields (4) and (5) can be set only together.

(1) IODON

Iodine pool model activation flag, 1 or 'ON', 0 or 'OFF'.

(type = integer / character*3, default = 0, units = none)

(2) IRCOPT

Set of chemical equations to use.

This parameter selects the set of chemical equations to use in the aqueous pool chemistry. The only set that has been tested is the default INSPECT-Powers set, and the others should be regarded as placeholders for the present.

(a) 0 or INSPECT-P
INSPECT-Powers set.

(b) 1 or INSPECT
INSPECT.

(c) 2 or BOYDBoyd
Carter and Dixon.

(d) 3 or MINIMUM
Minimum iodine set.

(type = integer / character*9, default=0, units = none)

(3) IODFLG

Iodine pool chemistry activation flag. Optional field.

This parameter is used to select between having the pool chemistry activate in a control volume only when iodine is present (the default behavior) and having the pool chemistry activate whether or not iodine is present, as long as the other activation criteria are satisfied. This allows calculation of water hydrolysis species without requiring iodine in the pool.

(a) 0 or IODINEPRESENT
Pool chemistry activates only with iodine present (default).

(b) 1 or OTHERCONDITIONS
Chemistry activates if other conditions are satisfied.

(type = integer / character*16, default = 0, units = none)

(4) IPCOEF1

Partition coefficient flag for I^0. Optional field.

This coefficient is used to switch the partition coefficient for I^0 on and off. The default is off (no atomic iodine released from the pool to the atmosphere).

(a) 1 or 'ON'
(b) 0 or 'OFF'

(type = integer / character*3, default = 0, units = none)

(5) IPCOEF2

Partition coefficient flag for HOI. 1 or ON, 0 or OFF. Optional field.

This coefficient is used to switch the partition coefficient for HOI on and off. The default is off (no HOI released from the pool to the atmosphere).

(type = integer / character*3, default = 0, units = none)

Note that use of the iodine pool model requires definition of at least 4 additional RN classes in the DCH module, which must be named as follows:

CH_3I – Methyl iodine
HCl – Hydrochloric acid
HNO_3 – Nitric acid
IM – Pool iodine bound in chemical species other than iodine or CsI.

If pool buffering is desired, two buffer classes may also be defined: HBO2, boric acid, and HPO4, sodium triphosphate. A third buffer class is recognized but not used currently: LIBO, lithium borate. Including the effect of silver in binding pool, iodine will require defining the silver iodide class, AGI. The additional classes must be named as given for the pool model to recognize them. Also, methane, carbon dioxide, hydrogen, and carbon monoxide must be defined in the *NCG* input.

Example

```
RN1_IOD01 ON  INSPECT-P  OTHERCONDITIONS  ON  ON
```

RN1_IOD02 – Iodine pool model start time

Optional.

The iodine pool chemistry model does not start until the problem time reaches TIOBEG. Calculation of atmospheric radiolysis to produce nitric and hydrochloric acid is not affected by the pool model start time and begins at the problem start time. This card may be used to delay startup of the pool chemistry model until later times in a plant accident scenario, where iodine is of most interest. The pool chemistry

solver can take a significant amount of CPU time, so this is another reason to delay pool startup.

(1) TIOBEG

Iodine pool model start time.

(type = real, default = -1.0E30, units = sec)

Example

```
RN1_IOD02  63.0E+3 ! Iodine pool model start time
```

RN1_IODCV – Iodine pool model Control Volume

Optional.

The iodine pool model (IPM) is active in the CVs given in the following table. All other CVs do not have any part of the IPM in use.

(1) NCV

Number of CVs in which iodine pool model is active.

(type = integer, default = none, units = none)

The following data are input as a table with length NCV:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) CVNAME

Control volume name for CV in which iodine pool model is active.

Example

```
RN1_IODCV  2 ! Iodine pool model active in 2 CVs
           1  CV100-Sump
           2  CV120-DryWell
```

RN1_PH – pH Calculation Types

Optional.

This record selects the time-dependent input source for the pool pH in a control volume, or alternatively the pH is calculated internally if no RN1_PH record is specified.

(1) NUMPHS

Length of pH calculation table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMPHS:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) NAMECVH

Control volume name containing the pool this data applies to.

(type = character*16, default = none, units = none)

(3) PHSRC

pH input source, which can be a table, control function, or external data file. The allowed values for PHSRC are

(a) 1 or TF

Source from table function SRCNAME.

(b) 2 or CF

Source from control function SRCNAME.

(c) 3 or EDF

Source from channel NUMCHAN of EDF SRCNAME.

(4) SRCNAME

Name of a table, control function, or external data file (EDF) (depends on PHSRC above).

(type = character*16, default = none, units = none)

(5) NUMCHAN

Channel number of EDF (required only if PHSRC = EDF or 3).

(type = integer, default = none, units = none)

Example

```
RN1_PH 1 ! N  CV NAME    PH SOURCE  CF/TF/EDF NAME  CHANNEL
           1  CVolume80      TF         TF577
```

RN1_DOS – Dose input

Optional.

This record is used to specify the dose rate to the atmosphere, pool, wall, and electric cable in a control volume. These rates will be used in the relevant parts of the iodine pool model to calculate radiolysis reactions. The atmosphere dose rate is used in homogeneous iodine reactions and nitric acid formation, the pool dose rate

applies to aqueous chemistry, the wall dose rate is used in surface reactions of iodine and methyl iodine, and the cable dose rate is used for generation of hydrochloric acid (by release of chlorine from plastics).

(1) NUMDOS

Length of dose input table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMDOS. Each data string consists of from four to seven the following fields:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) NAMECVH

Control volume name this data applies to.

(type = character*16, default = none, units = none)

(3) DOSID

Identifies site for dose rate.

(a) 1 or ATMO
Atmospheric dose rate.

(b) 2 or POOL
Pool dose rate.

(c) 3 or WALL
Wall dose rate.

(d) 4 or CABL
Cable dose rate.

(type = integer / character*4, default = none, units = none)

(4) DOSSRC

Source for time-dependent dose rate (Grays/s).

(a) 1 or OFF
Dose rate is 0.

(b) 2 or DC
Source is decay heat.

(c) 3 or TF
Source from table function SRCNAME.

(d) 4 or CF
Source from control function SRCNAME.

(e) 5 or EDF

Source from channel NUMCHAN of EDF SRCNAME.

(type = integer / character*3, default = none, units = none)

(5) SRCNAME

Name of a table, control function, or external data file (depends on DOSSRC above).

(type = character*16, default = none, units = none)

(5a) NUMCHAN

Channel number of EDF (required only if DOSSRC = EDF or 5).

(type = integer, default = none, units = none)

(6) SCALE

Optional scale factor that multiplies the radiation dose source. Optional field.

(type = real, default = 1.0, units = none)

Example

```
RN1_DOS 2 ! N CV NAME SITE SOURCE CF/TF/EDF NAME [CHANNEL] SCALE
            1 CVolume94   ATMO       TF     TF578           1.0
            2 CVolume94   POOL       CF     CF578           5.0
```

RN1_CB – Cable mass input

Optional.

This record specifies the cable mass in a control volume to be used in calculating the release of chlorine from plastics as a result of the radiation dose rate.

(1) NUMCAB

Length of cable mass input table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMCAB:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) NAMECVH

Control volume name.

(type = character*16, default = none, units = none)

(3) CABM

Cable mass in control volume.

(type = real, default = 0.0, units = kg).

Example

```
RN1_CB   1  ! N CV NAME          CABLE MASS
              1 CVolume94           234.5
```

RN1_WSC – Surface coating on walls

Optional.

If this record is omitted, the heat structure surface will be assumed to be the same as the material specified for the surface heat structure node.

(1) NUMWSC

Number of user-input surface coatings

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMWSC:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) IDS

Heat structure name.

(type = character*16, default = none, units = none)

(3) SIDE

Surface side.

(a) -1 or LHS
Left hand side.

(b) 1 or RHS
Right hand side.

(type = integer / character, default = none, units = none)

(4) COAT

Surface coating

(a) 0 or NONE
No coating, surface same as surface material.

(b) 1 or PAINT
Painted surface.

(c) 2 or STEEL
Steel surface.

(d) 3 or CONCRETE
Concrete surface.

(type = integer / character*8, default = none, units = none)

Example

```
RN1_WSC  5 ! N  HS NAME       SURFACE  COATING
             1  HS11011       LHS      STEEL
             2  HS10001       LHS      STEEL
             3  HS11001       RHS      STEEL
             4  HS11002       RHS      STEEL
             5  HS11003       RHS      STEEL
```

RN1_IOP – Iodine pool species to be output on plot file.

Optional.

This record specifies the aqueous pool species available for plotting on the plot file.

(1) NIODOT
Length of iodine pool species table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NIODOT:

(1) N
Data string number.

(type = integer, default = none, units = none)

(2) SPCNAM
The species name from the following list. The maximum total number of species that can be output is currently 10.

(type = character*8, default = none, units = none)

Table 3.1 Aqueous Species Names

E-	OH0	H0	H2O2	HO2-	HO20
O-	O2-	O3-	I2-	I3-	I0
HOI-	I2AQ	I2OH-	HOI0	OI-	IO0
I2O2	HIO20	IO2-	IO20	HIO3-	IO32-
HOI30	IO3-	IO30	HI-	FE2+	CH3
CH3I	CH3I+	CH4I0	CH2O	CH3O	CH3OH
FE3+	H+	H2AQ	OH-	O2AQ	I-

Example

```
RN1_IOP   6 ! N       SPECIES NAME
```

```
1          HOI-
2          I2AQ
3          I2OH-
4          HOI0
5          OI-
6          IO0
```

3.1.12 Flashing-Jet Model

RN1_JI – Impacting jet model

Optional.

This record turns on the flashing-jet, impact model for a heat structure. There must be either a flowpath with the flashing model turned on (see FL Package User's Guide) or a set of sources to the target control volume to provide the water droplet aerosol distribution.

(1) NUMJI

A number of heat structures for which flashing-jet impact model is on.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMJI:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) HSNAM

Heat structure user name.

(type = character*16, default = none, units = none)

(3) SIDE

Heat structure side:

(a) -1 or LHS
Left hand side.

(b) 1 or RHS
Right hand side.

(type = character*3 / integer, default = none, units = none)

(4) AJET

Jet entrance area.

(type = real, default = none, units = m^2)

(5) CVNAM

Control volume user name for target CV.

(type = character*16, default = none, units = none)

(6) KSP

Key for source of pressure.

(a) 1 or CFCF

Providing source pressure, next field should be (7a).

(b) 0 or CONST

Constant source pressure, next field should be (7b).

(type = character*5 / integer, default = none, units = none)

optionally followed by

(7a) CFNAM

Control function name for CF providing source pressure. This field is required if and only if KSP = -1 or KSP = CF.

(type = character*16, default = none, units = none)

or

(7b) SPVALUE

Constant source pressure. This field is required if and only if KSP = 0 or KSP = CONST.

(type = real, default = none, units = Pa)

(8) S1

Distance from jet equilibrium point to plate (HS).

(type = real, default = none, units = m)

(9) CFNAM1

Control function user name for mass flow of water.

(type = character*16, default = none, units = none)

Example

```
RN1_JI   1 ! N HSNAME SIDE AJET CVNAME Key Name/Value S1 CFNAME1
             1  HS1   LHS  3.0   CV5   CF  CF5        5.  CF6
```

In the model implementation, the steam-water masses in the target CV are used to calculate the mixture specific volume; hence, this volume should be sized to represent the conditions at the equilibrium point after flashing.

<u>Model Limitations</u>

There are a few limitations on the model due to implementation. First, there can only be one jet impinging on a given heat structure, although the same jet can impinge on more than one heat structure. Second, since the mixture-specific volume is obtained from the water-steam masses in the target volume, there is an initial delay in this parameter converging to its correct value. The initial delay can be minimized by sizing the target volume correctly; i.e., do not make it too large.

3.1.13 Plate Film Entrainment Model

RN1_ENT – Entrainment from films on HS surfaces

Optional.

This record calculates entrainment from films on HS surfaces.

(1) LenTable

Number of entrainment models, dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length LenTable:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) HSNAME

Heat structure name.

(type = character*16, default = none, units = none)

(3) SIDE

Heat structure side to calculate film entrainment.

(a) -1 or LHS
Left side.

(b) 1 or RHS
Right side.

(type=integer / character*3, default = none, units = none)

(4) CFVEL

Control function name for CF providing fluid velocity.

(type = character*16, default = none, units = none)

(5) CFDIAM

Control function name for CF providing entrainment drop diameter or a constant diameter value.

(type = character*16 / real, default = none, units = m)

(6) XMULT

Multiplier on entrainment rate. Optional.

(type = real, default = 1.0, units = none)

3.1.14 Lower Head Molten Pool Model

RN1_MPCR – Molten pool concentration ratios of radionuclides

Optional.

This record specifies the partitioning concentration ratio of radionuclides between the ceramic molten pool and the metallic molten pool for each radionuclide class. The partitioning concentration ratio is the ratio of concentrations between the ceramic molten pool and the metallic molten pool (i.e., C_1/C_2 where C_1 is the concentration in the ceramic molten pool and C_2 is the concentration in the metallic molten pool). A user-defined control function name or model name (in this version only the name DEFAULT is allowed) may be entered for each class. If no record exists for a class, the default model is used. Currently the default model is an equal concentration model (C_1/C_2 = 1), which calculates the same radionuclide concentration for both the ceramic molten pool and the metallic molten pool.

(1) LenTable

Number of radionuclide classes for which the partitioning concentration ratio is defined by user; dimension of the table below.

(type = integer, default = 0, units = none)

The following data are input as a table with length LenTable:

(1) N

Data string index.

(type = integer, default = none, units = none)

(2) CLNAME

Radionuclide class name.

(type = character*16, default = none, units = none)

(3) MODEL TYPE

Radionuclide partitioning concentration ratio model type.

(a) -1 or CF

Use control function.

(b) 0 or DEFAULT

Use the default model.

(type=integer / character*7, default = DEFAULT, units = none)

(4) CFNAME

User-defined control function name.

(type = character*16, default = none, units = none)

Example

```
RN1_MPCR   2 ! Number RN Class Name  Model name  CF name
  1         BA          CF         CF72
  2         I2          DEFAULT
```

3.2 MELCOR Input

The MELCOR input available for the RN package is as follows:

RN1_EDTFLG – RN Edit Flags

Optional.

User controls for RN text edits

(1) IFLGDM

Edit flag for deposited RN masses:

(a) 0 or BYPASS
Bypasses the edits.

(b) 1 or PRINT
Prints the edits.

(type = integer / character*6, default = 0, units = none)

(2) IFLGGM

Edit flag for gas RN masses:

(a) 0 or BYPASS
Bypasses the edits.

(b) 1 or PRINT
Prints the edits.

(type = integer / character*6, default = 0, units = none)

(3) IFLGLM

Edit flag for liquid RN masses:

(a) 0 or BYPASS
Bypasses the edits.

(b) 1 or PRINT
Prints the edits.

(type = integer / character*6, default = 0, units = none)

(4) IFLGCA

Edit flag for chemisorption masses:

(a) 0 or BYPASS
Bypasses the edits.

(b) 1 or PRINT
Prints the edits.

(type = integer / character*6, default = 0, units = none)

(5) IFLGI2

Edit flag for iodine pool model:

(a) 0 or BYPASS
Bypasses the edits.

(b) 1 or PRINT
Prints the edits.

(type = integer / character*6, default = 0, units = none)

Example

```
RN1_EDTFLG   1   1   1   0   0
```

RN1_IOD02 – Iodine pool model start time

Optional.

The iodine pool chemistry model does not start until the problem time reaches TIOBEG. Calculation of atmospheric radiolysis to produce nitric and hydrochloric acid is not affected by the pool model start time and begins at the problem start time. This card may be used to delay startup of the pool chemistry model until later times in a plant accident scenario, where iodine is of most interest. The pool chemistry solver can take a significant amount of CPU time, so this is another reason to delay pool startup.

(1) TIOBEG

Iodine pool model start time.

(type = real, default = -1.0E30, units = sec)

Example

```
RN1_IOD01   600.
```

RN1_VISUAL – Special-purpose output for visualization

Optional.

This record allows creation of a special-purpose output file for visualization of aerosol and vapor behavior in volumes. If present, a file will be created for each volume in the list, and each will contain data for aerosol and vapor inventories associated with the volume.

By default, the file names will be RN1VISDATA(NAMECVH).txt. The string 'RN1VISDATA' can be replaced by input of RN1VISUALFILE in the environmental data block.

(1) NUMVIS

Length of volume visualization table.

(type = integer, default = 0, units = none)

The following data are input as a table with length NUMVIS. Each data string consists of from four to seven the following fields:

(1) N

Data string number.

(type = integer, default = none, units = none)

(2) NAMECVH

Control volume name for which output is desired.

(type = character*16, default = none, units = none)

Example

```
RN1_VISUAL   2
1   CV100
2   CV200
```

4 Sensitivity Coefficients

There are two types of sensitivity coefficients in the RN1 package: ordinary SC and RN class SC. The input of ordinary SC is performed in standard manner by the RN1_SC record. As regards the RN class SC, their input is performed in a peculiar manner. This is caused by the following reason. In MELCOR 2.0, the RN class is specified by the name, not the number. So the user should specify the RN class name (see the example of RN class SC input after description of RN1_CSC record) while setting SC for the given RN class. As a consequence, the dimension of the RN class SC decreases by 1 in comparison with MELCOR 1.8.6.

The names of the default RN classes for MELCOR 2.1 are

XE, CS, BA, I2, TE, RU, MO, CE, LA, UO2, CD, AG, BO2, H2O, CON, CSI, and CSM.

If a user specifies a new RN class, s/he should use its name as the input of the SC belonging to this class. The input of RN class SC is performed by RN1_CSC record.

There are the following RN class sensitivity coefficients in the RN1 package:

(1) Scalar SC: 7103, 7130, 7136, 7141

(2) One-dimensional SC: 7102, 7107, 7111, 7120, 7131, 7132, 7170

(3) Two-dimensional SC: 7101, 7110

Finally, note that sensitivity coefficient arrays 7130, 7131, 7132, 7134, 7140, 7141, 7142, 7143, and 7144 described below pertain to metallic fuels and are not applicable to the LWR version of MELCOR. The models are not described in the RN Package Reference Manual.

7000 – Differential Equation Convergence

These values are used to control the solution of the aerosol dynamics equations. At each timestep, the equations are solved by one of two available methods, depending on the rate of change of the sectional mass distribution. If the rate is not excessive, then the distribution is updated using the start-of-step derivatives in an explicit Euler step. For larger rates, the equations are integrated by the RKF45 routine developed at Sandia National Laboratories.

(1)` RKF45 absolute error tolerance
(default = 1.0E-18, units = none, equiv = none)

(2) RKF45 relative error tolerance, and relative tolerance for conservation of component masses in RN.
(default = 1.0E-3, units = none, equiv = none)

(3) Maximum fractional decrease in any sectional density permitted during an explicit step.
(default = 0.1, units = none, equiv = none)

(4) Maximum fractional increase in any sectional density permitted during an explicit step.
(default = 0.1, units = none, equiv = none)

(5) Sectional density below which the fractional change restrictions are ignored.
(default = 1.0E-12, units = kg/m^3, equiv = none)

7001 – Aerosol Coefficient Criteria

These values are used to control the evaluation of the aerosol dynamics coefficients. These values are used to determine if the numerical integration performed is satisfactory.

(1) Aerosol coefficient absolute error,
(default = 1.0E-18, units = none, equiv = none)

(2) Aerosol coefficient relative error,
(default = 1.0E-3, units = none, equiv = none)

7002 – Fission Product Decay Beta Range

This value defines the range of characteristic beta radiation from fission product decay and is used to modify the fraction of decay heat that is deposited in a control volume atmosphere. *SPVOL1* (records RN1_DHV) and *SPSUR2* (records RN1_DHS) are multiplied by a factor

min [1.0, Atmosphere_density x CVPATH / C7002(1)],

where *CVPATH* is the characteristic path in the volume (see description of RN1_DHL table record). Any reduced deposition is compensated by proportionate increase in energy distributed to other components specified by *SPVOL2-SPVOL4* or *SPSUR3-SPSUR5*. (The calculation is bypassed if the sum of these other split coefficients is zero.)

(1) Characteristic range of beta radiation fission product decay,
(default = 1.2, units = kg/m^2, equiv = *RANGE*)

7003 – Parameters for Interpolation of Aerosol Coefficients

For a given pressure and temperature, values of the aerosol coefficients are interpolated from values precalculated for the (P, T) points defined on the RN1_PT input record. The interpolation is bilinear in P^{PEXP} and T^{TEXP}.

(1) Power of temperature in the interpolation.
(default = 0.5, units = none, equiv = *TEXP*)

(2) Power of pressure in the interpolation.
(default = -1.0, units = none, equiv = *PEXP*)

7100 – COR Material Release Multipliers

These values are used to scale the nominal release rates obtained from the CORSOR release models described below. All values must be between 0.0 and 1.0, inclusive. By default the scale factor is unity for the fuel material and zero for all other core materials because the CORSOR models are strictly applicable to the fuel material.

(1)` Fuel material multiplier
(default = 1.0, units = none, equiv = none)

(2) Structural Zr multiplier
(default = 0.0, units = none, equiv = none)

(3) Oxidized Zr multiplier
(default = 0.0, units = none, equiv = none)

(4) Structural steel multiplier
(default = 0.0, units = none, equiv = none)

(5) Oxidized steel multiplier
(default = 0.0, units = none, equiv = none)

(6) Control rod poison multiplier
(default = 0.0, units = none, equiv = none)

(7) Structural Inconel multiplier
(default = 1.0E-12, units = kg/m3, equiv = none)

7101 – CORSOR Coefficients

These values are used to determine the release of aerosols and vapors from the fuel in the core during heating. They vary with RN material class. Default values of these coefficients are the generally accepted CORSOR values. The release rate is

Release Rate (fraction / min) = A exp (BT)

The coefficients are in C7101(i,j) where

i – Location index, dimensioned 3
j – Type of parameter,
= 1, Temperature value, °C
= 2, CORSOR coefficient A value, fraction/min
= 3, CORSOR coefficient B value, $°C^{-1}$

The interpretation of the values is as follows:

For a temperature between C7101(i,1) and C7101(i + 1,1), the CORSOR A and B coefficients C7101(i,2) and C7101(i,3) are used for the releases. For temperature values less than C7101(1,1), no releases are evaluated. For temperatures greater

than C7101(3,1), the values C7101(3,2) and C7101(3,3) are used. The default values of the coefficients are

	C7101(1,1)	C7101(2,1)	C7101(3,1)
All default Classes except the TE Class	900.E0	1400.E0	2200.E0
TE Class	900.E0	1600.E0	2000.E0

The A and B values are

Class		C7101(1,j)	C7101(2,j)	C7101(3,j)
XE	A	7.02E-9	2.02E-7	1.74E-5
	B	0.00886	0.00667	0.00460
CS	A	7.53E-12	2.02E-7	1.74E-5
	B	0.0142	0.00667	0.00460
BA	A	7.50E-14	8.26E-9	1.38E-5
	B	0.0144	0.00631	0.00290
I2	A	7.02E-9	2.02E-7	1.74E-5
	B	0.00886	0.00667	0.00460
TE	A	1.62E-11	9.04E-8	6.02E-6
	B	0.0106	0.00552	0.00312
RU	A	1.36E-11	1.36E-11	1.40E-6
	B	0.00768	0.00768	0.00248
MO	A	5.01E-12	5.93E-8	3.70E-5
	B	0.0115	0.00523	0.00200
CE	A	6.64E-12	6.64E-12	1.48E-7
	B	0.00631	0.00631	0.00177
LA	A	5.00E-13	5.00E-13	5.00E-13
	B	0.00768	0.00768	0.00768
UO2	A	5.00E-13	5.00E-13	5.00E-13
	B	0.00768	0.00768	0.00768
CD	A	1.90E-12	5.88E-9	2.56E-6
	B	0.0128	0.00708	0.00426
AG	A	1.90E-12	5.88E-9	2.56E-6
	B	0.0128	0.00708	0.00426
BO2, H2O, CON, CSI, CSM	A	0.	0.	0.
	B	0.	0.	0.
User-defined RN Classes	A	0.	0.	0.
	B	0.	0.	0.

The input of C7101(:,:) is performed by RN1_CSC record.

7102 – CORSOR-M Coefficients

The values of the CORSOR-M coefficients from CORSOR-M are in this record series. This model uses an Arrhenius form of the release equation, or

$$\textit{Release Rate (fraction / min)} = k_o \exp(-Q / RT)$$

The values in the expression designated k_o, Q, and T are in units of min^{-1}, kcal/mol, and K, respectively. The value of R is 1.987E-3 in the appropriate units. The default values in MELCOR for each class are

Class	k_o (C7102(1))(1./min)	Q (C7102(2))(kcal/mole)
XE	2.00E5	63.8
CS	2.00E5	63.8
BA	2.95E5	100.2
I2	2.00E5	63.8
TE	2.00E5	63.8
RU	1.62E6	152.8
MO *	23.15*	44.1*
CE	2.67E8	188.2
LA **	1.46E7 **	143.1**
UO2	1.46E7	143.1
CD **	5.95E3**	70.8**
AG	5.95E3	70.8
BO2, H2O, CON, CSI, CSM	0.	0.
User-defined RN Classes	0.	0.

Note. The CORSOR-M model does not consider release from Mo Class , La Class or Cd Class to be significant. Previous versions of MELCOR used zero values for these classes when using CORSOR-M. In MELCOR 2.0 non-zero release coefficients are supplied as described.

* Coefficients for CORSOR-M Mo Class are based on a curve fit to the CORSOR release model for Mo Class.

** Coefficients for CORSOR-M La Class are set identical to the CORSOR-M UO2 Class values, following the same assumption as used in the CORSOR model for Mo Class. Likewise for Cd and Ag Classes.

The input of C7102(:) is performed by the RN1_CSC record.

7103 – CORSOR-Booth Class Scaling Factors: Nominal Values

The release rate for species other than *cesium* is given by multiplying the cesium release rate by an appropriate scaling factor for each RN class:

$$\textit{Release Rate(Class)} = \textit{Release Rate(Cs)} \times \textit{C7103(Class)}$$

Class	C7103	Class	C7103
XE	1.0	LA	4.0E-8

Class	C7103	Class	C7103
CS	1.0	UO2	3.2E-4
BA	4.0E-4	CD	0.25
I2	0.64	AG	0.16
TE	0.64	BO2	0.0
RU	2.5E-3	H2O	0.0
MO	1.0E-3	CON	0.0
CE	4.0E-8	CSI	0.64
CSM	1.0	User-defined RN Classes	0.0

C7103 is a scalar for every class. The input of C7103 is performed by the RN1_CSC record.

7104 – Release Surface-to-Volume Ratio

This value is the base value for the CORSOR and CORSOR-M releases when the surface-to-volume ratio option is specified. The rates are modified as follows:

$$\textit{Release Rate} = \textit{Release Rate (CORSOR or CORSOR-M)} \times (S/V)_{structure} / (S/V)_{base}$$

where the $(S/V)_{base}$ is derived from the CORSOR test data.

(1) Surface-to-volume base ratio.
(default = 4.225E2, units = m^{-1}, equiv = none)

7105 – Modification of Release Rates

These values are used to modify the CORSOR or CORSOR-M coefficients as determined by conditions other than used in the release expressions. For example, the release of tellurium is affected by the amount of cladding oxidation (see Section 2.3.1 of the RN Package Reference Manual). At present, this mechanism is the only one addressed in this sensitivity coefficient series. In this case, the CORSOR or CORSOR-M release rate is used when the amount of cladding oxidation is greater than the cut-off value or when there is no cladding present. When the amount of cladding oxidation is less than the cut-off value, the release rate is multiplied by the multiplier given below. The values below will be used if the user sets radionuclide class name TE only. The default values are

(1) Cladding oxidation fraction cut-off point.
(default = 0.70, units = none, equiv = none)

(2) Release rate multiplier (Used when amount of oxidation is less than the cut-off value).

(default = 1/40 = 0.025, units = none, equiv = none)

7106 – CORSOR-Booth Transient Release Parameters for Cesium

The classical or effective diffusion coefficient for cesium in the fuel matrix is given by

$$D = D0 \times EXP(-Q / RT)$$

where R is the universal gas constant, T is the temperature, Q is the activation energy and $D0$ is the pre-exponential factor given by this sensitivity coefficient below.

The CORSOR-Booth method gives the cesium release fraction as a function of

D x TIME / A^2, where A is the equivalent sphere radius for the fuel grain. Two sets of coefficients are defined. The first, i=1, is used for normal fuel. The second, i=2, is for an alternate fuel. It is used only if an alternate fuel and associated classes are defined by input on the RN1_BOOTH00 and RN1_BOOTH01 input records. The default values are identical to those for normal fuel.

(1,i)	Low burn-up value of *D0* for fuel type i. (default = 2.3E-9, units = m2/s, equiv = none)
(2,i)	High burn-up value of *D0* for fuel type i. (default = 2.3E-9, units = m^2/s, equiv = none)
(3,i)	Burn-up value at which value of *D0* changes for fuel type i. (default = 3.E4, units = MWD/MTU, equiv = none)
(4,i)	Activation energy *Q* for fuel type i. (default = 2.411E5, units = J/kg-mole, equiv = none)
(5,i)	Equivalent sphere radius of fuel grain for fuel type i. (default = 6.0E-6, units = m, equiv = none)

7107 – CORSOR-Booth Class Scaling Factors: Oxidation Modified

The scaling factors given by C7103 above must be modified under certain conditions of cladding oxidation.

When the oxide mass fraction exceeds C7107(1) and the temperature (TEMP) exceeds C7107(2), the class scaling factor is given by,

$$SFACT = C7107(3) \times EXP(C7107(4) \times MIN(TEMP , C7107(5)))$$

When the oxide mass fraction is below C7107(6) the class scaling factor is given by,

SFACT = C7107(7)

Class	C7107(i)						
	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7
XE, CS	1.1	0.0	0.0	0.0	0.0	-1.0	0.0
BA	1.1	0.0	0.0	0.0	0.0	5.0E-01	2.0E-03
I2	1.1	0.0	0.0	0.0	0.0	-1.0	0.0
TE	1.1	0.0	0.0	0.0	0.0	7.0E-01	6.4E-01
RU	7.5E-01	2.3E+03	2.5E-03	0.0	2.7E+03	-1.0	0.0
MO, CE	1.1	0.0	0.0	0.0	0.0	-1.0	0.0
LA	1.1	0.0	0.0	0.0	0.0	5.0E-02	4.0E-8
UO2	1.1	0.0	0.0	0.0	0.0	-1.0	0.0
CD	7.5E-01	2.0E+03	2.5E-01	0.0	2.3E+03	-1.0	0.0
AG	7.5E-01	2.0E+03	1.6E-01	0.0	2.3E+03	-1.0	0.0
BO2, H2O, CON, CSI, CSM	1.1	0.0	0.0	0.0	0.0	-1.0	0.0
User-defined RN Classes	1.1	0.0	0.0	0.0	0.0	-1.0	0.0

The input of C7107(:) is performed by the RN1_CSC record.

7110 – Vapor Pressure

The vapor pressure curves for the fission product vapors are included in these sensitivity coefficients. These values determine the amount of fission product vapor released from the core and the amount condensed onto the heat structures and aerosol particles. These coefficients give the vapor pressure through the following relationship:

$$\log_{10} [\text{Pressure (mm Hg)}] = -A / T + B + C \log_{10} (T)$$

The values of A, B, and C are stored in the C7110(i,j) array where

i - Location index, dimensioned 3
j - Type of parameter,

= 1, Temperature value, K
= 2, Coefficient A
= 3, Coefficient B
= 4, Coefficient C

The interpretation of these values is as follows:

For a temperature value C7110(1,1) equal to -1.E0, the class is always an aerosol, or, in other words, the vapor pressure is 0. It is not recommended that users define classes that are radioactive (i.e., generate decay heat) to be always an aerosol because they may overheat surfaces on which they are deposited. Even the most refractory of materials will vaporize and discontinue heating a surface when the temperature becomes high enough (above 4000K). If the first coefficient A value, C7110(1,2) is equal to -1.E0, the class is always a vapor. For other values, if the temperature is between the i and i+1 temperature values, the coefficients at i are used. For temperatures lower than the first temperature value, the vapor pressure is 0. For temperatures higher than the last temperature value, the last range coefficient values are extrapolated and used. The values presently in the code are given below. They are based on preliminary data in (see the RN Reference Manual), but values flagged by '*' have been adjusted slightly to give a continuous variation of vapor pressure as a function of temperature. Classes BA and RU, MO, CE, LA, UO2, CD, AG, and BO2 now have vapor pressure curves; classes H2O and CON have the same values, which are representative of a nonvolatile (refractory) oxide (the values are actually those for UO_2 above 3000K). Hence, these classes will not vaporize until the temperature exceeds 3000K.

Class		C7110(1,j)	C7110(2,j)	C7110(3,j)
XE	T	0.	10000.	-1.
	A	-1.E0 (always a vapor)		
CS	T	600.	1229.5	-1.
	A	1.36E4	1.21E4	
	B	8.895	7.675	
	C	0.0	0.0	
BA	T	1000.	-1.	-1.
	A	1.10E4		
	B	8.4		
	C	0.0		
I2	T	273.	387.	457.
	A	3578.	3205.	2176.912045
	B	17.72	23.66536*	7.637352*
	C	-2.51	-5.18	0.0
TE	T	273.	-1.	-1.
	A	13940.		
	B	23.51		
	C	-3.52		
RU	T	1500.	-1.	-1.

Class		C7110(1,j)	C7110(2,j)	C7110(3,j)
	A	33200.		
	B	11.6088		
	C	0.0		
MO	T	1500.	-1.	-1.
	A	32800.		
	B	9.68		
	C	0.0		
CE	T	1500.	-1.	-1.
	A	21570.		
	B	8.74		
	C	0.0		
LA	T	1500.	-1.	-1.
	A	21800.		
	B	8.683		
	C	0.0		
UO2	T	1500.	-1.	-1.
	A	32110.		
	B	11.873		
	C	0.0		
CD	T	1000.	-1.	-1.
	A	13730.		
	B	8.43		
	C	0.0		
AG	T	1000.	-1.	-1.
	A	15400.		
	B	8.15		
	C	0.0		
BO2	T	1000.	-1.	-1.
	A	19520.		
	B	11.125		
	C	0.0		
H2O, CON	T	3000.	-1.	-1.
	A	18000.		
	B	8.875		
	C	0.0		
CSI	T	600.	894.	1553.
	A	10420.	9678.	7303.903158*
	B	19.70	20.34569*	7.58405103*
	C	-3.02	-3.52	0.0
CSM	T	600.	1229.5	-1.
	A	13600.	12100.	
	B	8.895	7.675	
	C	0.0	0.0	

Class		C7110(1,j)	C7110(2,j)	C7110(3,j)
User-defined RN Classes	T	0.	10000.	
	A	-1.E0 (always a vapor).		

The input of C7110(:,:) is performed by the RN1_CSC record.

7111 – Vapor Diffusivity Constants

These values are used to determine the fission product vapor diffusivity. The diffusivity values are used in the determination of the transport of condensed vapors to the walls and aerosols in each volume. The values in 7111 are for the RN material class masses. For each class, two different values are specified. They are

C7111(1) = *Sigma*, Angstroms

C7111(2) = *E/K*, Deg K

The sigma and E/K values are Lennard-Jones parameters where sigma is a characteristic diameter of the molecule and E/K is the characteristic energy of interaction between the molecules divided by the Boltzmann constant. The default values of these parameters are

Class	Sigma (C7111(1)) (Angstroms)	E/K (C7111(2)) (K)
XE	4.055	229.
CS, BA	3.617	97.
I2	4.982	550.
TE, RU, …, CSI, CSM	3.617	97.
User-defined RN Classes	3.617	97.

Most of the classes are defaulted to water vapor values due to the lack of information.

The input of C7111(:) is performed by the RN1_CSC record.

(Sensitivity coefficients 7112 and 7115, previously used in calculation of diffusivities of RN vapors through atmosphere gases, have been eliminated. The calculation now uses modeling and data in the Materials Properties [MP] package.)

7120 – Class Molecular Weights

Two class molecular weights are specified by these coefficients. The first value represents the molecular weight as the species exists in the fuel (typically an

elemental value), which will be used in the class combination model to determine the total number of moles released that are available for combination with other classes. The second value represents a compound molecular weight (if data are available) after the species reacts with a nonradioactive mass (oxygen or water) when it is released. For example, the Cs class might have an elemental mass equal to Cs and a compound mass equal to CsOH. The compound molecular weight is used in the diffusivity calculations and in the reaction mass transfer calculations. These values are stored as follows:

C7120(1) = Molecular Weight – Elemental

C7120(2) = Molecular Weight – Compound

The units are kg/kg-mole. The default values in the code are

Class	MW-E [C7120(1)]		MW-C [C7120(2)]	
XE	Xe	131.3	Xe	131.3
CS	Cs	132.905	CsOH	149.913
BA	Ba	137.34	Ba	137.34
I2	I_2	253.8008	I_2	253.8008
TE	Te	127.6	TeO	143.6
RU	Ru	101.07	Ru	101.07
MO	Mo	95.94	Mo	95.94
CE	Ce	140.12	Ce	140.12
LA	La	138.91	La	138.91
UO2	U	238.03	UO_2	270.03
CD	Cd	112.4	Cd	112.4
AG	Sn	118.69	Sn	118.69
BO2	B_2O_3	69.622	B_2O_3	69.622
H2O	H_2O	18.016	H_2O	18.016
CON	conc.	28.97	conc.	28.97
CSI	CsI	259.8054	CsI	259.8054
CSM	CsM	361.75	CsM	425.75
User-defined RN Classes	conc.	28.97	conc.	28.97

The input of C7120(:) is performed by the RN1_CSC record.

7130 – Oxidation-Based Release Coefficients for Metallic Fuels

When a metallic fuel oxidizes, a substantial change occurs in the structure of the fuel matrix. This change allows many fission products to escape almost instantaneously

as the fuel is oxidized. The oxidation-based release model assumes that the release rate of each fission product is proportional to the rate of fuel oxidation. Thus, the release equation takes the form

$$I = k\,F$$

where I is the integrated release fraction, F is the fraction of the fuel that is oxidized, and k is a proportionality constant specified in this coefficient. This model is applied to each core cell and fuel element independently in order to properly account for varying fission product concentrations throughout the core. The default values in MELCOR for each class are

Class	k [C7130]	Class	k [C7130]
XE	1.00	CE	0.01
CS	0.30	LA	0.01
BA	0.01	UO2	0.01
I2	1.00	CD	0.01
TE	1.00	AG	0.01
RU	0.05	BO2,H2O, CON, CSI, CSM	0.00
MO	0.01	User-defined RN Classes	0.00

C7130 is a scalar for every class. The input of C7130 is performed by the RN1_CSC record.

This model is not used with the LWR COR package.

7131 – Birney Non-Oxidation Release Coefficients for Metallic Fuel

When metallic fuels melt in a non-oxidizing environment, the release of fission products is gradual over time and dependent upon the vapor pressure of the fission product class. The release correlation takes the form

$$dM/dt = -R\,M_o \text{ where } R = k\,(P/P_o)^v$$

where M_o is the initial mass, R is the Birney release rate (fraction/minute), k is a proportionality constant (fraction/minute), v is a normalization constant (none) for the ratio of partial pressures, and P and P_o are the vapor pressure of the fission product at the current temperature and the reference temperature (1473 K), respectively, as calculated using the form in sensitivity coefficient 7110. The values of k, v, and a maximum allowed release rate R_m are specified here in coefficient 7113. This model

is applied to each core cell and fuel element independently to properly account for varying fission product concentration throughout the core. The default values in MELCOR are

Class	k[C7131(1)] (1./min)	v [C7131(2)] (none)	R_m[C7131(3)] (1./min)
XE	0.050	0.70	1.0E5
CS	0.00048	0.70	1.0E5
BA	0.00010	0.70	1.0E5
I2	0.00320	0.70	1.0E5
TE	0.00170	0.70	1.0E5
RU	0.00010	0.70	1.0E5
MO	0.00010	0.70	1.0E5
CE	0.00010	0.70	1.0E5
LA	000010	0.70	1.0E5
UO2	2.50E-6	0.70	1.0E5
CD	0.00010	0.70	1.0E5
AG	0.00010	0.70	1.0E5
BO2,H2O, CON, CSI, CSM	0.00	0.70	1.0E5
User-defined RN Classes	0.00	0.70	1.0E5

The input of C7131(:) is performed by the RN1_CSC record.

This model is not used with the LWR COR package.

7132 – Arrhenius Non-Oxidation Release Coefficients for Metallic Fuel

When metallic fuels melt in a non-oxidizing environment, the release of fission products is gradual over time and dependent upon the vapor pressure of the fission product class. The release correlation takes the form

$$dM/dt = -R\, M_o \text{ where } R = k_o\, exp(\, -Q/R_{gas}\, T)$$

where R is the Arrhenius release rate (fraction/minute), k_o is a proportionality constant (fraction/minute), R_{gas} is the universal gas constant in units of kcal/(mole K), Q is a correlation constant in units of kcal/mole, and T is the fuel temperature (K). The values of k_o, Q, and a maximum allowed release rate R_m are specified in this coefficient. This model is applied to each core cell and fuel element independently in order to properly account for varying fission product concentration throughout the core. The default values in MELCOR for each class are

Class	k_o [C7132(1)] (1./min)	Q [C7132(2)] (kcal/mole)	R_m [C7132(3)] (1./min)

Class	k_o [C7132(1)] (1./min)	Q [C7132(2)] (kcal/mole)	R_m [C7132(3)] (1./min)
XE	0.00	0.00	1.0E5
CS	0.00	0.00	1.0E5
BA	0.00	0.00	1.0E5
I2	0.00	0.00	1.0E5
TE	0.00	0.00	1.0E5
RU	0.00	0.00	1.0E5
MO	0.00	0.00	1.0E5
CE	0.00	0.00	1.0E5
LA	0.00	0.00	1.0E5
UO2	0.00	0.00	1.0E5
CD	0.00	0.00	1.0E5
AG	0.00	0.00	1.0E5
BO2,H2O, CON, CSI, CSM	0.00	0.00	1.0E5
User-defined RN Classes	0.00	0.00	1.0E5

The input of C7132(:) is performed by the RN1_CSC record.

This model is not used with the LWR COR package.

7135 – Noble Gas Release on Failure of Metallic Fuel

Metallic fuel elements normally fail through cladding rupture. When the cladding ruptures, much of the noble gas inventory is released in a puff. The failure itself is calculated in the COR Package, but the noble gas release is calculated in the RN package using this coefficient. The values below will not be used if the user doesn't set class XE (noble gas):

(1) Fraction of the noble gas inventory in the fuel element and cell that is released on fuel failure. The remainder is left in the fuel to be released using the time-dependent release models.

(default = 0.80, units = none, equiv = none)

(2) Fraction of the released gas that is placed in the center channel when the inner fuel fails. The remainder is placed in the inner annulus.

(default = 0.50, units = none, equiv = none)

(3) Fraction of the released gas that is placed in the inner annulus when the outer fuel fails. The remainder is placed in the outer annulus.

(default = 0.50, units = none, equiv = none)

This model is not used with the LWR COR package.

7136 – Solubility of RN Classes in Water Films

If part or all of a water film drains from a surface of a heat structure to the pool in the associated control volume, any fission products deposited on that surface are normally relocated with the water in proportion to the fraction of the film that is drained. These coefficients allow the film fission product relocation behavior to be modified by changing the fraction of fission product deposits that are assumed to be dissolved in—and, therefore, relocate with—the film.

C7136 is the fraction of the class deposited on a surface that is treated as dissolved in any water film on that surface. Thus

fraction_of_class_relocated = C7136 x fraction_of_film_relocated.

Class	C7136
XE, …, CSM	1.0
User-defined RN Classes	1.0

C7136 is a scalar for every class. The input of C7136 is performed by the RN1_CSC record.

7140 – Release from Molten U-Al Pools

These values are used to alter the release rates of fission products from pools of molten U-Al. The model is described in more detail in the pool release model reference.

This model is not used with the LWR COR package.

(1) Contact angle of an escaping bubble with the pool surface.
(default = 170.0, units = degrees, equiv = none)

(2) The contact angle of a nucleating bubble with the nucleation site at bubble departure. The departure diameter is determined from

$$Departure_diameter = 0.0208 \times C7140(2) \times \sqrt{(\sigma / g)}$$

(default = 10.0, units = degrees, equiv = none)

(3) Coefficient of bubble coalescence and breakup.

$$n_{bub}/t = C7140(3) \times r^3_{bub} \times n^2_{bub} + nucleation - rise$$

(default = 1.0, units = none, equiv = none)

(4) Minimum permitted bubble diameter.

(default = 1.0E-4, units = m, equivalence = none)

(5) Mean size of nucleation cavity openings.

(default = 1.0E-4, units = m, equiv = none)

(6) Standard deviation of normal distribution of nucleation cavity openings.

(default = 3.6E-5, units = m, equiv = none)

(7) Maximum concentration of nucleation sites present in the pool for a temperature between the solidus and liquidus temperatures of the fuel.

(default = 6.5E6, units = number/kg, equiv = none)

(8) Concentration of permanent nucleation sites existing at all temperatures.

(default = 100.0, units = number/kg, equiv = none)

(9) Coefficient in correlation for the Sherwood (Nusselt) number used to calculate mass transfer at the pool surface.

$Sh = C7140(9)\ Ra^{C(7140(10))}$, Ra = Rayleigh number.

(default = 0.2, units = none, equiv = none)

(10) Exponent in above correlation.

(default = 0.26, units = none, equiv = none)

(11) Fraction (between 0.0 and 1.0) of the pool surface open for collection of bubbles.

(default = 1.0, units = none, equiv = none)

(12) Diffusivity of all RN classes in the molten metal pool.

(default = 1.0E-11, units = m^2/s, equiv = none)

(13) Switch defining limits on nucleation of fission products.

= 1.0 No limits; fission product material is immediately available to nucleation site.

= 2.0 Diffusion limited; nucleation is limited by the rate at which fission products diffuse to nucleation sites.

= 3.0 Convection and diffusion limited; nucleation is limited by the rate at which fission products convect and diffuse to nucleation sites.

(default = 1.0, units = none, equiv = none)

7141 – Solubility of Classes in Al-U Alloy

Solubility of each fission product class in a molten metal pool, used as the concentration below which it is assumed that there is no driving force present for the

release of that class. That is, for lower concentrations there will be no release of the class by the pool release model.

This model is not used with the LWR COR package.

(1) Solubility of class (for all default and user-defined classes).
(default = 0.0, units = moles/liter, equiv = none)

C7141 is a scalar for every class. The input of C7141 is performed by the RN1_CSC record.

7142 – Debris Particle of Average Surface Area

This dimension defines the surface-to-volume ratio for debris. This model is not used with the LWR COR package.

(1) Dimension of particle with average surface area.
(default = 3.2E-4, units = m, equiv = none)

7143 – Molten Fraction Criterion for Release from U-Al Pools

Below this mole fraction, the CORSOR or CORSOR-M model of fission product release will be applied. At higher molten fractions, the fission product release calculation will switch to the molten pool model.

This model is not used with the LWR COR package.

(1) Molten fraction at which release calculation switches to the molten pool model.
(default = 0.62, units = none, equiv = none)

7144 – Temperature Criterion for Release from Intact Fuel

No release will be calculated from intact fuel below this temperature.

This model is not used with the LWR COR package.

(1) Threshold temperature for FP release from intact fuel.
(default = 933.0, units = K, equiv = none)

7150 – SPARC-90 Model Parameters

These parameters are used to control the SPARC-90 Pool Scrubbing model.

(1) Number of spatial steps taken in tracking the ascent of the bubbles as they rise through the pool.

(default = 10., units = none, equiv = *XNRISE*)

(2) Number of angular steps taken between 0 and 180 degrees in calculating heat and mass transfer from the interior to the surface of the rising bubbles.

(default = 5., units = none, equiv = *XNCIRC*)

(3) Error tolerance in the calculation of the saturation ratio in the rising bubbles.

(default = 1.E-4, units = none, equiv = *ERRSMX*)

(4) Maximum number of iterations permitted during the calculation of the saturation ratio.

(default = 25., units = none, equiv = *XITSMX*)

(5) Error tolerance in the calculation of the energy transfer in the rising bubbles.

(default = 1.E-4, units = none, equiv = *ERREMX*)

(6) Maximum number of iterations permitted during the calculation of the energy transfer in the rising bubbles.

(default = 25., units = none, equiv = *XITEMX*)

(7) Error tolerance in the calculation of the temperature in the rising bubbles.

(default = 1.E-3, units = none, equiv = *ERRTMX*)

(8) Maximum number of iterations permitted during the calculation of the temperature in the rising bubbles.

(default = 25., units = none, equiv = *XITTMX*)

(9) Maximum decontamination factor. Calculated decontamination factors that exceed this value will result in the complete removal (i.e., absorption into the pool) of the associated vapor or aerosol.

(default = 1.E12, units = none, equiv = *DFMAX*)

(10) Vent exit condensation decontamination factor scaling factor. This factor is applied to the decontamination factor that is calculated as a result of steam condensation in the vent exit region that occurs when the bubbles are thermally equilibrated with the pool temperature. GE's Moody suggested a value of 3. at one time according to the authors of SPARC-90.

(default = 1.0, units = none, equiv = *ECMULT*)

7151 – SPARC-90 Globule Size Correlation

This correlation relates the initial size of the globule formed to the Weber number of the gas exiting the vent with velocity V_o. The cube of the initial diameter of the gas globule is given by

$$D_g^{\,3} = 15\, v_n\, D_o^2 \left(\sigma / \rho_l\, g\right)^{1/2}$$

where D_o is the equivalent diameter of the vent, σ and ρ_l are the surface tension and density of the pool and the normalized volume v_n is given by

$$v_n = a\, \mathrm{We}^b \quad (\mathrm{We} = \rho_l D_o V_o^2 / \sigma)$$

and the constants a and b are given by constants C7151(1,i) and C7151(2,i), respectively for vent type i = MVENT.

i (vent type)	C7151(1,i)	C7151(2,i)
1 (sparger)	3.45	0.46
2 (downcomer)	0.0891	0.616
3 (horizontal)	0.857	0.73

7152 – SPARC-90 Bubble Size/Shape Model

The volume mean diameter of bubbles formed at the vent exit is given by

$$d_{vm} = C7152(1) \cdot 10^{C7152(2) + \left(C7152(3) + C7152(4) \cdot x_{nc}\right)^{C7152(5)}}$$

where x_{nc} is the noncondensible gas fraction in the gas exiting the vent. The bubbles are modeled as oblate spheroids with a major to minor axis ratio given by

$$\frac{a}{b} = C7152(6) + C7152(7)\, d_{vm} + C7152(8)\; d_{vm}^2$$

(1) Initial bubble diameter correlation coefficient.
(default = 7.E-3, units = m, equiv = *DIAMI_C7152*)

(2) Initial bubble diameter correlation exponent constant.
(default = -0.2265, units = none, equiv = none)

(3) Initial bubble diameter correlation exponent constant.
(default = 0.0203, units = none, equiv = none)

(4) Initial bubble diameter correlation exponent constant.
(default = 0.0313, units = none, equiv = none)

(5) Initial bubble diameter correlation exponent constant.
(default = 0.5, units = none, equiv = none)

(6) Oblate spheroid major/minor axis ratio correlation constant.
(default = .84107, units = none, equiv = none)

(7) Oblate spheroid major/minor axis ratio correlation constant.
(default = 1.13466, units = cm^{-1}, equiv = none)

(8) Oblate spheroid major/minor axis ratio correlation constant.
(default = -0.3795. units = cm^{-2}, equiv = none)

7153 – SPARC-90 Bubble Rise Velocity Model

The bubble rise velocity relative to the liquid is given by the following correlation:

$$V_r = C7153(1)\,(\sigma / \rho_l)^{1/4} \text{ (cm/s) for } d_{vm} \le 0.5 \text{ cm}$$

$$V_r = C7153(3)\,V_r(d_{vm} = 0.5 \text{ cm})\, d_{vm}^{C7153(4)} \text{ (cm/s) otherwise}$$

where σ and ρ_l are the surface tension and density of the liquid, respectively.

(1) Coefficient for rise velocity correlation of small bubbles.
(default = 7.876, units = cm/s, equiv = none)

(2) Transition diameter from small to large bubble rise velocity correlations.
(default = 0.5, units = cm, equiv = none)

(3) Coefficient for rise velocity correlation of large bubbles.
(default = 1.40713, units = none, equiv = none)

(4) Exponent for diameter in rise velocity correlation of large bubbles.
(default = 0.49275, units = none, equiv = none)

7154 – SPARC-90 Swarm Velocity Model

The bubble swarm velocity correlation is given by

$$\overline{V}_s = \text{MIN}\left\{ 0.5\left[V_s(x=0) + V_s(x=h_p)\right] C7154(5)\right\}$$

where

$$V_s(x) = \left\{\left[(Q_s + C7154(1))\right] / C7154(2)\right\}^{C7154(3)} \left[1 - C7154(4)x\right] (cm/s)$$

x is the vertical distance from the surface of the pool and h_p is the depth of the pool at the vent exit.

(1)	Constant in the swarm velocity correlation. (default = 5.33, units = liter/s, equiv = none)
(2)	Constant in the swarm velocity correlation. (default = 3.011E-3, units = liter-s/cm^2, equiv = none)
(3)	Exponent in the swarm velocity correlation. (default = 0.5, units = none, equiv = none)
(4)	Coefficient of x in equation for $V_s(x)$. (default = -3.975E-4, units = cm^{-1}, equiv = none)
(5)	Maximum permitted value of the swarm velocity. (default = 170., units = cm/s, equiv = *VSWRMX*)

7155 – SPARC-90 Particle Impaction Model

If gas leaves the vent exit at a high velocity, the initial globules rapidly lose that velocity. The forward globular interface, as it slows and stops, can capture particles if they have sufficient inertia. Inertia and drag of particle size i is represented by the Stokes number

$$Stk_i = \frac{\rho_i V_e d_i^2}{9\,\mu\, D_0}$$

where

d_i	= Particle diameter
ρ_i	= Particle density
V_e	= Vent exit gas velocity (before equilibration with pool)
μ	= Gas viscosity
D_o	= Vent exit orifice diameter

The decontamination factor for this impaction process is given by

$$DF_{II,i} = (1-\alpha_i)^{-1}$$

where

$$\alpha_i = C7155(1)\cdot C7155(2)^{C7155(3)\sqrt{Stk_i}} \text{ if } \sqrt{Stk_i} \le C7155(4)$$

$$\alpha_i = C7155(5)\cdot C7155(6)^{C7155(7)\sqrt{Stk_i}} \text{ otherwise}$$

and α_i is limited to a maximum value of C7155(8).

(1) Constant in α_i is the correlation for small Stokes number.
(default = 1.79182, units = none, equiv = none)

(2) Constant in α_i is the correlation for small Stokes number.
(default = 3.3437E-11, units = none, equiv = none)

(3) Constant in α_i is the correlation for small Stokes number.
(default = 5.9244E-3, units = none, equiv = none)

(4) Transition value of SQRT(Stk_i) for changing from small to large Stokes number correlation for α_i.
(default = 0.65868, units = none, equiv = none)

(5) Constant in α_i correlation for a large Stokes number.
(default = 1.13893, units = none, equiv = none)

(6) Constant in α_i correlation for a large Stokes number.
(default = 1.4173E-6, units = none, equiv = none)

(7) Constant in α_i correlation for a large Stokes number.
(default = 4.2597E-3, units = none, equiv = none)

(8) Maximum permitted value of α_i.
(default = 0.99, units = none, equiv = none)

7156 – SPARC-90 Solute Ionization Correlations

The van't Hoff ionization factors are used in modeling hygroscopic effects that promote steam condensation on hygroscopic aerosol particles even in subsaturated atmospheres. The temperature-dependent correlations have the form

$I_{sol}[T°C] = I_{sol}[C7156(9)][1 + C7156(8)\{T-C7156(9)\}]$ (sol = CsOH or CsI)

$I_{CsOH}[C7156(9)] = C7156(6) + C7156(7)[n_{CsOH}/n_T]$

$I_{CsI}[C7156(9)] = C7156(1) + C7156(2)[n_{CsI}/n_T]$ for $n_{CsI}/n_T \leq C7156(3)$

$I_{CsI}[C7156(9)] = C7156(4) + C7156(5)[n_{CsI}/n_T]$ otherwise

$I_{sol'}[T°C] = C7156(10)$ for all other solutes in the droplet

where $n_T = n_{CsOH} + n_{CsI} + n_{sol} + n_w$ is the total number of moles and $n_{sol'}$ is the number of moles of soluble material excluding CsOH and CsI in the growing droplet/aerosol.

(1) Additive constant in a low concentration CsI correlation.
(default = 1.79417, units = none, equiv = none)

(2) Multiplicative constant in a low concentration CsI correlation.
(default = -3.34363, units = none, equiv = none)

(3) Transition CsI molar ratio between CsI correlations.
(default = 0.021, units = none, equiv = none)

(4) Additive constant in a high concentration CsI correlation.
(default = 1.63439, units = none, equiv = none)

(5) Multiplicative constant in a high concentration CsI correlation.
(default = 4.30022, units = none, equiv = none)

(6) Additive constant in a CsOH correlation.
(default = 1.75467, units = none, equiv = none)

(7) Multiplicative constant in a CsOH correlation.
(default = 20.7974, units = none, equiv = none)

(8) Multiplicative constant in a temperature correction correlation.
(default = -0.002321, units = none, equiv = none)

(9) Reference temperature for CsI and CsOH van't Hoff ionization correlations.
(default = 25., units = °C, equiv = none)

(10) Temperature independent van't Hoff ionization constant for other solutes.
(default = 2., units = none, equiv = XIVH)

7157 – SPARC-90 Settling Velocity Correlation

A set of empirical correlations are used to determine the Reynolds number, Re, from which $V_{g,i}$, the settling velocity of particles of size *i* in the rising bubbles, follows as

$$V_{g,i} = (\mu\, Re)/(\rho_g d_i)$$

where μ and ρ_g are the viscosity and density of the gas, respectively, and d_i is the diameter of the particle. The empirical correlations are

$$f\,(\mathrm{Re}) = \left(4\,\rho_i\,\rho_g\,g\,d_i^3\right)/\left(3\,\mu^2\right)$$

Re = $[f(Re)/C7157(2)]^{C7157(3)}$ if $C7157(1) < f(Re) < C7157(4)$
= $[f(Re)/C7157(5)]^{C7157(6)}$ if $C7157(4) \le f(Re) < C7157(7)$
= $[f(Re)/C7157(8)]^{C7157(9)}$ if $C7157(7) \le f(Re) < C7157(10)$
= $[f(Re)/C7157(11)]^{C7157(12)}$ if $C7157(10) \le f(Re) < C7157(13)$
= $[f(Re)/C7157(14)]^{C7157(15)}$ if $C7157(13) \le f(Re)$

For particles with a diameter less than about 70μm, the gravitational settling velocity follows Stokes law and is given by

$$V_{g,i} = \left(\rho_i \, g \, S_i \, d_i^2\right) / \left(18\mu\right) \text{if } f(\text{Re}) \le C7157(1)$$

where S_i is the Cunningham slip correction factor for particles of size *i*.

(1) First bound on *f(Re)* in the settling velocity correlation.
(default = 9.6, units = none, equiv = none)

(2) First denominator in the settling velocity correlation.
(default = 27.00, units = none, equiv = none)

(3) First exponent in the settling velocity correlation.
(default = 1./1.130, units = none, equiv = none)

(4) Second bound on *f(Re)* in the settling velocity correlation.
(default = 93.6, units = none, equiv = none)

(5) Second denominator in the settling velocity correlation.
(default = 24.32, units = none, equiv = none)

(6) Second exponent in the settling velocity correlation.
(default = 1./1.227, units = none, equiv = none)

(7) Third bound on *f(Re)* in the settling velocity correlation.
(default = 410., units = none, equiv = none)

(8) Third denominator in the settling velocity correlation.
(default = 15.71, units = none, equiv = none)

(9) Third exponent in the settling velocity correlation.
(default = 1./1.417, units = none, equiv = none)

(10) Fourth bound on *f(Re)* in the settling velocity correlation.
(default = 1.07E4, units = none, equiv = none)

(11) Fourth denominator in the settling velocity correlation.
(default = 6.477, units = none, equiv = none)

(12) Fourth exponent in the settling velocity correlation.
(default = 1./1.609, units = none, equiv = none)

(13) Fifth bound on *f(Re)* in the settling velocity correlation.
(default = 2.45E5, units = none, equiv = none)

(14) Fifth denominator in the settling velocity correlation.
(default = 1.194, units = none, equiv = none)

(15) Fifth exponent in the settling velocity correlation.
(default = 1./1.867, units = none, equiv = none)

7158 – SPARC-90 HOI Correlation

The partition coefficient for organic iodine (CH_3I) is given by the following correlation:

$$H_{OI} = T / 10^{C7158(1)/T + C7158(2)}$$

with the temperature, T, given in degrees K.

(1) First constant in exponent of correlation for H_{OI}.
(default = -1388.89, units = K, equiv = none)

(2) Second constant in exponent of correlation for H_{OI}.
(default = 6.461, units = none, equiv = none)

7159 – SPARC-90 I2 Chemistry Model Parameters

The first eleven parameters are used to calculate the temperature-dependent equilibrium constants for the five chemical reactions considered in the I_2 scrubbing model. The reactions considered are

$$I_{2(g)} \leftrightarrow I_{2(aq)}$$
$$I_{2(aq)} + I^- \leftrightarrow I_3^-$$
$$I_{2(aq)} + H_2O \leftrightarrow H^+ + I^- + HIO$$
$$I_{2(aq)} + H_2O \leftrightarrow H_2OI^+ + I^-$$
$$H_2O \leftrightarrow H^+ + OH^-$$

The equilibrium constant for the ith reaction (i = 1,..,4) is given by

$$EQK(i) = C7159(2 \cdot i - 1)\exp[C7159(2*i)/T]$$

while the equilibrium constant for the dissociation of water is given by

$$EQK(5) = C7159(9)\exp[C7159(10)/T + C7159(11)/T^2]$$

The twelfth parameter is a value of gaseous iodine concentration (moles/cm^3) that is used to indicate when saturation of the pool prevents continued iodine scrubbing. If the iodine vapor concentration in the bubbles exceeds this value but scrubbing is prevented by a large iodine concentration in the pool, then a message is issued to notify the user of the condition. The thirteenth parameter is the relative error tolerance used in iteratively solving the system of iodine chemistry equilibrium equations for the iodine partition coefficient.

(1) Coefficient of the equilibrium constant for the first iodine chemistry reaction.
(default = 1.3882E-3, units = none, equiv = *CGITST*)

(2) Constant used in the exponent of the equilibrium constant for the first iodine chemistry reaction.

(default = 3279.3, units = K, equiv = none)

(3) Coefficient of the equilibrium constant for the second iodine chemistry reaction.

(default = 7.7606, units = $moles^{-1}$, equiv = none)

(4) Constant used in the exponent of the equilibrium constant for the second iodine chemistry reaction.

(default = 1370., units = K, equiv = none)

(5) Coefficient of the equilibrium constant for the third iodine chemistry reaction.

(default = 1.0423E-2, units = $moles^2$, equiv = none)

(6) Constant used in the exponent of the equilibrium constant for the third iodine chemistry reaction.

(default = -7148., units = K, equiv = none)

(7) Coefficient of the equilibrium constant for the fourth iodine chemistry reaction.

(default = 4.2271E-9, units = moles, equiv = none)

(8) Constant used in the exponent of the equilibrium constant for the fourth iodine chemistry reaction.

(default = -1748.5, units = K, equiv = none)

(9) Coefficient used in the equilibrium constant for the dissociation of water.

(default = 1.56531E-13, units = $moles^2$, equiv = none)

(10) First constant used in the exponent of the dissociation constant for water.

(default = 5462.81, units = K, equiv = none)

(11) Second constant used in the exponent of the dissociation constant for water.

(default = -1.87376E6, units = K^2, equiv = none)

(12) Minimum iodine vapor concentration required to issue a message indicating the cessation of scrubbing due to pool saturation.

(default = 1.E-6, units = $moles/cm^3$, equiv = none)

(13) Relative error tolerance used in solving the iodine equilibrium equations.

(default = 1.E-3, units = none, equiv = *ERRTOL_C7159*)

7160 – Chemisorption

The coefficients give the chemisorption rate for species *j* on surface type *i* through the following equation:

$$k_{ij} = a_{ij} e^{\frac{-E_{ij}}{RT_i}}$$

The values of *a* and *E* are stored in the array C7160(i,j) where

i - Type of parameter,
= 1, chemisorption coefficient a, m/s
= 2, activation energy E, J/kg
j - CA class, dimensioned *NUMCA* (the user-defined number of CA classes).

The default values in MELCOR for each chemisorption class are

Class	a[(C7160(1,j)] (m/s)	E [C7160(2,j)] (J/kg)	Interaction
1	0.139	5.96E7	CsOH - Stainless Steel
2	0.035	5.96E7	CsOH – Inconel
3	2.0E-7	0.0	CsI - Stainless Steel
4	2.0E-7	0.0	CsI – Inconel
5	5.5E-7	2.49E7	HI - Stainless Steel
6	9.0E-10	3.39E7	I2 - Stainless Steel
7- NUMCA	0.0	0.0	-

The input of C7160(:,:) is performed by the RN1_SC record, using two indices.

7170 – Hygroscopic Aerosol Parameters

The user-adjustable parameters for the hygroscopic aerosol model contained in coefficient array C7170 are described in the following paragraphs. The coefficients allow the user to (1) input the water solubility of RN class materials as a linear function of temperature, (2) adjust the ionization factor for the RN material (basically the number of ions formed upon dissolving in water), and (3) input/adjust the material density of the RN class compound.

The solubility in water of an RN class N can be specified at two temperatures, between which the solubility is assumed to vary linearly. Outside of this temperature range, the solubility is held constant at the value associated with the temperature range boundary value. Usually the temperature range is specified in reference books as cold water, taken to be 273K, and hot water, taken to be 373K. The user can adjust these temperature limits. The material solubility is expressed in units of kg of solute per kg of <u>water</u> (*note, not kg solution*). Upon being dissolved in water the RN

class compound is assumed to form two (2) ions, as would be the case for CsI $\square$ Cs + I^-. The user can modify this to some other value to represent a more complex dissolution process. In order to assure smoothness in the model behavior as higher temperature is approached (eg. approaching the water critical point), the RN class solubility is gradually reduced to zero over a temperature range, normally between 600K and 647K). The solubility or ionization factor is also reduced to zero over this same temperature range.

For each RN class material the following C7170 coefficients can be adjusted. The coefficient array is dimensioned once for each RN class, where the index is the specific coefficient value.

(1) Reference temperature for low temperature saturation solubility.
(default = 273.0, units = K , equiv = none).

(2) Reference temperature for high temperature saturation solubility.
(default = 373.0, units = K , equiv = none).

(3) Saturation solubility at low temperature reference.
(default =See Table 4.1, units = [kg/kg H_2O], equiv = none).

(4) Saturation solubility [kg/kg H_2O] at high temperature reference.
(default = See Table 4.1, units = [kg/kg H_2O], equiv = none).

(5) Temperature to begin linearly reducing solubility factor.
(default = 600.0, units = K, equiv = none).

(6) Temperature above which solubility factor is zero.
(default = 647.0, units = K, equiv = none).

(7) Value of ionization factor below C7170(5).
(default = 2.0, units = dimensionless , equiv = none).

(8) Value of ionization factor above C7170(6). The default value is 0.
(default = 0.0, units = dimensionless , equiv = none).

(9) Class compound material density.
(default = See Table 4.1, units = [kg/m^3], equiv = none).

The following table provides those default values for C7170 that are unique to a specific RN class. The user may also want to modify the RN class compound molecular weight when modeling specific aerosol materials that are not default RN class compounds, such as NaOH or MgO. These are accessible through sensitivity coefficient array C7120(2).

Table 4.1 Default Hygroscopic Properties of RN Class Compounds.

RN Class	**Compound Density** [kg/m^3]	**Compound Solubility At 273K** [kg/kg H_2O]	**Compound Solubility At 373K** [kg/kg H_2O]	**Ionization Factor**
Sensitivity Coefficient	C7170(9)	C7170(3)	C7170(4)	C7170(7)
XE - Noble Gases: Xe	1	0	0	--
CS - Alkali Metals: CsOH	3675	3.95	3.95	2
BA - Alkaline Earths: BaO	5720	0	0	2
I2 - Halogens: I_2	1	0	0	2
TE - Chalcogens: Te	5680	0	0	2
RU - Platinoids: Ru	6970	0	0	2
MO - Transition: Mo	7470	0	0	2
CE - Tetravalent: Ce	7000	0	0	2
LA - Trivalents: La	6510	0	0	2
UO2 - Uranium: UO2	10960	0	0	2
CD - Cd	8150	0	0	2
AG - Sn	6446	0	0	2
BO2 - Boron	2520	0	0	2
H2O - Water: H2O	1000	--	--	0
CON - Concrete	2250	0	0	2
CSI – CsI	4510	.44	2.25	2
CSM – CsM	4030	.67	.67	2

The input of C7170(:) is performed by the RN1_CSC record.

7180 – Iodine Pool Model Mass Transfer Parameters

These mass transfer parameters are used to calculate the pool-to-atmosphere and atmosphere-to-surface transfer rates, and the nitric/hydrochloric acid atmosphere-to-surface deposition rates. Default values for pool-atmosphere and iodine-steel wall give good agreement with the ISP-41 test problem. The values for iodine-painted walls are unknown, and the defaults are the same as for the steel wall.

(1) Pool-atmosphere mass transfer rate.
(default = 0.003, units = m/s, equiv = none).

(2) HCl acid deposition rate.
(default = 0.003, units = m/s, equiv = none).

(3) H_2NO_3 acid deposition rate.
(default = 0.003, units = m/s, equiv = none).

(4) Iodine-steel wall adsorption rate.
(default = 0.0009, units = m/s, equiv = none).

(5) Iodine-steel wall desorption rate.
(default = 9E-7, units = 1/s, equiv = none).

(6) Iodine-painted wall adsorption rate.
(default = 0.0009, units = m/s, equiv = none).

(7) Iodine-painted wall desorption rate.
(default = 9E-7, units = 1/s, equiv = none).

7181 – Iodine Pool Chemistry Iteration Parameters

These are used to control the solution of the iodine pool chemistry equations. The defaults give good convergence on the ISP-41 test problem. The difference between parameter (3) and parameters (4) and (5) is that parameter (3) is applied to each chemical species after solution by the chemistry solver; parameters (4) and (5) are passed to the solver to determine a local convergence for each species.

(1) ODE chemistry solver effective timestep.
(default = 1.0, units = s, equiv = none).

(2) ODE chemistry solver maximum number of timesteps.
(default = 4000, units = none, equiv = none).

(3) Chemistry solver species-relative error tolerance.
(default = 0.0001, units = none, equiv = none).

(4) Chemistry solver absolute convergence error.
(default = 1E-13, units = kmole/m^3, equiv = none).

(5) Chemistry solver relative convergence error.
(default = 0.0001, units = none, equiv = none).

7182 – Iodine Pool Chemistry Activation Limits

These are used to define when the pool chemistry solution is activated. These are used because the chemistry solution has a limited range of validity. The parameters are consistent with late-time conditions in a plant accident scenario.

(1) Minimum iodine concentration in pool.
(default = 1E-10, units = kmole/m^3, equiv = none).

(2) Maximum atmosphere pressure.
(default = 1E6, units = Pa, equiv = none).

(3) Minimum atmosphere volume.
(default = 0.1, units = m^3, equiv = none).

(4) Maximum atmosphere temperature.
(default = 500, units = K, equiv = none).

(5) Maximum pool temperature.
(default = 425, units = K, equiv = none).

(6) Minimum pool volume.
(default = 0.0, units = m^3, equiv = none).

RN1_SC – RN1 Sensitivity Coefficients Input

Optional.

This record allows users to change the default RN1 sensitivity coefficient values to user-input ones. This record is not required.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the RN1 sensitivity coefficient array.
(type = integer, default = none, units = none)

(3) VALUE
New value of the sensitivity coefficient. Values must be real-valued only – no integer values are allowed.
(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA
First sensitivity coefficient index.
(type = integer, default = none, units = none)

(5) NB
Second sensitivity coefficient index. Specific sensitivity coefficient dependent.
(type = integer, default = none, units = none)

Example:

```
RN1_SC  5  ! N   SCnumber     Value       Index1 Index2
             1   7000         0.2E-11       5
```

```
2   7151        0.9          2      3
3   7159        1.5E-2       13
4   7160        0.6E8        2      1
5   7182        0.45E3       4
```

RN1_CSC – RN Class Sensitivity Coefficients Input

Optional.

This record allows users to change the default RN class sensitivity coefficient values to user-input ones. This record is not required.

(1) NUMSTR

Number of data strings.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Data string index.

(type = integer, default = none, units = none)

(2) NNNN

Unique four-digit identifier of the RN class sensitivity coefficient array.

(type = integer, default = none, units = none)

(3) CLASS

RN class name.

(type = character*16, default = none, units = none)

(4) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(5) NA

First sensitivity coefficient index. Specific sensitivity coefficient dependent.

(type = integer, default = none, units = none)

(6) NB

Second sensitivity coefficient index. Specific sensitivity coefficient dependent.

(type = integer, default = none, units = none)

Example:

```
RN1_CSC  4  ! N  SCnumber  ClassName  Value   Index1 Index2
              1  7101         I2      1.8E-4    3      2
              2  7101         MO      0.13E-1   1      3
              3  7102         RU      0.15E3    2
              4  7103         CE      0.35E-4
```

5 Plot Variables

The RN package variables that may be used for plot variables are listed and described below. Note that control function arguments (some that are identical in definition to these plot variables but different in format) are described in the following section.

> **Note**: Those variables that are not described in this section but are described in the following (control function arguments) can be plotted by inclusion of EXEC_PLOT input record in MELGEN and MELCOR input. See the EXEC Package Users' Guide for details. While inconvenient, this was found necessary to avoid generation of excessively large plot files.

RN1-CPUC	Total time for the run routines of the RN1 package. (units = s)
RN1-CPUE	Total time for the edit routines of the RN1 package. (units = s)
RN1-CPUR	Total time for the restart routines of the RN1 package. (units = s)
RN1-CPUT	Sum of the run, edit, and restart times of the RN1 package. (units = s)
RN1-ATMG.cv	Total mass of aerosol (radioactive plus nonradioactive) in the gas phase, for each control volume index cv. (units = kg)
RN1-ARMG.cv	Total mass of radioactive aerosol in the gas phase, for each control volume index *cv*. (units = kg)

RN1-VTMG.cv	Total mass of fission product vapor (radioactive plus nonradioactive) in the gas phase, for each control volume index *cv*. (units = kg)
RN1-VRMG.cv	Total mass of radioactive fission product vapor in the gas phase, for each control volume index *cv*. (units = kg)
RN1-ATML.cv	Total mass of aerosol (radioactive plus nonradioactive) in the liquid phase, for each control volume index *cv*. (units = kg)
RN1-ARML.cv	Total mass of radioactive aerosol in the liquid phase, for each control volume index *cv*. (units = kg)
RN1-VTML.cv	Total mass of fission product vapor (radioactive plus nonradioactive) in the liquid phase, for each control volume index *cv*. (units = kg)
RN1-VRML.cv	Total mass of radioactive fission product vapor in the liquid phase, for each control volume index *cv*. (units = kg)
RN1-XMRLSE-x-y.cv	Total mass of class index *x* released from COR components or CAV debris in control volume index *cv*. The parameter *y* specifies total mass as a compound (y=1), the radioactive mass only (y=2), or the total mass as an element, including nonradioactive sources and/or structural releases (y=3). (units = kg)
RN1-XMRLSET	Total non-radioactive plus radioactive mass released from COR components. (units = kg)
RN1-XMRLSER	Total radioactive mass released from COR components. (units = kg)

RN1-TOTMAS-y.x	Sum of masses of class index *x* in all locations, excluding that not yet released from fuel or debris. The parameter *y* specifies either total mass (y=1) or just the radioactive mass (y=2). (units = kg)
RN1-ATMT	Total radioactive plus non-radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures. (units = kg)
RN1-ATMR	Total radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures. (units = kg)
RN1-VTMT	Total radioactive plus non-radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures. (units = kg)
RN1-VTMR	Total radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures. (units = kg)
RN1-TMT	Total radioactive and non-radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to *RN1-ATMT* plus *RN1-VTMT* variables. (units = kg)
RN1-TMR	Total radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to *RN1-ATMR* plus *RN1-VTMR* variables. (units = kg)
RN1-MDTT-n-y	Total radioactive and non-radioactive mass deposited on heat structure index *n* on side *y*. The values of *y* are 1 for the LHS and 2 for the RHS. (units = kg)

RN1-MDTR-n-y	Total radioactive mass deposited on heat structure index *n* on side *y*. The values of *y* are 1 for the LHS and 2 for the RHS. (units = kg)
RN1-TMDTT	Total radioactive and non-radioactive mass deposited on all heat structures. (units = kg)
RN1-TMDTR	Total radioactive mass deposited on all heat structures. (units = kg)
RN1-DHTOT	Total decay heat calculated by the RN package for all locations of radionuclides. (units = W)
RN1-DHCOR	Total decay heat from the radionuclides in the core. (units = W)
RN1-DHCAV	Total decay heat from the radionuclides in the cavity. (units = W)
RN1-DHDEP	Total decay heat from the radionuclides deposited on the heat structures. (units = W)
RN1-DHATM	Total decay heat from airborne radionuclides. (units = W)
RN1-DHPOL	Total decay heat from radionuclides in the pool. (units = W)
RN1-TYCLT-x-2.ty	Radioactive mass of class index *x* in all control volumes of type index *ty*, including mass deposited on heat structures associated with the control volumes. (units = kg)
RN1-TYTOT-2.ty	Sum of radioactive masses of radionuclides in all control volumes of type index *ty*. Includes mass in pool and atmosphere, but not that deposited on heat structures.

	(units = kg)
RN1-MMDW.cv	Mass median diameter of the wet aerosol distribution in the gas phase for each control volume index *cv*. (units = m)
RN1-GSDW.cv	Geometric standard deviation of the wet aerosol distribution in the gas phase for each control volume index *cv*. (units = none)
RN1-MMDD.cv	Mass median diameter of the dry aerosol distribution in the gas phase for each control volume index *cv*. (units = m)
RN1-GSDD.cv	Geometric standard deviation of the dry aerosol distribution in the gas phase for each control volume index *cv*. (units = none)
RN1-PH.nnn	pH of the pool in control volume index *nnn*. (units = none)
RN1-IOP-cccccc.nnn	Concentration of aqueous species name *cccccc* in control volume index *nnn*. The c means that the pool species must be specified on record RN1_IOP before it can be output. This is done because there are many pool species (39) and the user does not necessarily want them all on the plot file, increasing the size of the plot output file. The species *cccccc* can be one of the names in the table in the RN1_IOP record description. (units = kmole/m^3)
RN1-TMCAT	Total mass of chemisorbed species deposited on all heat structures. (units = kg)
RN1-TMCAR	Total radioactive mass of chemisorbed species deposited on all heat structures. (units = kg)

RN1-MCAT-y-n	Total mass chemisorbed on side *y* of heat structure index *n*. The LHS is y=1 and the RHS is y=2. (units = kg)
RN1-MCAR-y-n	Total radioactive mass chemisorbed on side *y* of heat structure index *n*. The LHS is y=1 and the RHS is y=2. (units = kg)
RN1-MMDC-x.cv	Mass median diameter of component *x* in the aerosol distribution in the gas phase for each control volume index *cv*. (units = m)
RN1-GSDC-x.cv	Geometric standard deviation of component *x* in the aerosol distribution in the gas phase for each control volume index *cv*. (units = none)
RN1-ENT-MDOT.en	Mass flux of entrainment for entrainment model index *en*. (units = kg/m^2-s)
RN2-CPUC	Total time for the run routines of the RN2 package. (units = s)
RN2-CPUE	Total time for the edit routines of the RN2 package. (units = s)
RN2-CPUR	Total time for the restart routines of the RN2 package. (units = s)
RN2-CPUT	Sum of the run, edit, and restart times of the RN2 package. (units = s)
RN2-AMFLT-x.f	Total aerosol mass of class index *x* on filter *f*, where *f* is the filter index. (units = kg)
RN2-RAFLT-x.f	Radioactive aerosol mass of class index *x* on filter index *f*.

	(units = kg)
RN2-AMFLT.f	Total aerosol mass on filter index *f* (sum of RN2-AMFLT-x.f) (units = kg)
RN2-VMFLT-x.f	Total vapor mass of class index *x* on filter index *f*. (units = kg)
RN2-RVFLT-x.f	Radioactive vapor mass of class *x* on filter *f*. (units = kg.)
RN2-VMFLT.f	Total vapor mass on filter index *f* (sum of RN2-VMFLT-x.f). (units = kg)
RN2-FLT-QTOT.f	Total decay heat from all radionuclides deposited on filter index *f*. (units = W)
RN2-FLT-QLOS.f	Heat loss from filter index *f* (portion of RN2-FLT-QTOT that is assumed to be lost from the system). (units = W)
RN2-VFLT-TMP.f	Temperature of charcoal bed in vapor filter index *f*. (units = K)
RN2-VFLT-RAD.f	Radiolytic desorption rate of fission product vapors from filter index *f*. (units = kg/s)
RN2-VFLT-THR.f	Thermal desorption rate of fission product vapors from filter index *f*. (units = kg/s)
RN2-VFLT-BUR.f	Rate of fission product vapor release from filter index *f* due to charcoal combustion. (units = kg/s)
RN2-DFBUB-ip-sec.w	Instantaneous decontamination factor of aerosols in mass section *w* from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume associated

with flow path index NFL, ip=10*NFL+2 for the *to* volume associated with flow path index NFL, and ip=10*NCAV for the pool associated with cavity index NCAV.)

RN2-DFBUB-ip-a

Instantaneous decontamination factor of total aerosol mass from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume associated with flow path NFL, ip=10*NFL+2 for the *to* volume associated with flow path NFL, and ip=10*NCAV for the pool associated with cavity NCAV.)

RN2-DFBUB-ip-cls.v

Instantaneous decontamination factor of vapor in class index *v* from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume associated with flow path index NFL, ip=10*NFL+2 for the *to* volume associated with flow path index NFL, and ip=10*NCAV for the pool associated with cavity index NCAV.)

RN2-DFBBT-ip-sec.w

Cumulative decontamination factor of aerosols in mass section *w* from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume associated with flow path index NFL, ip=10*NFL+2 for the *to* volume associated with flow path index NFL, and ip=10*NCAV for the pool associated with cavity index NCAV.)

RN2-DFBBT-ip-a

Cumulative decontamination factor of total aerosol mass from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume associated with flow path index NFL, ip=10*NFL+2 for the *to* volume associated with flow path index NFL, and ip=10*NCAV for the pool associated with cavity index NCAV.)

RN2-DFBBT-ip-cls.v

Cumulative decontamination factor of vapor in class index *v* from pool scrubbing associated with path *ip*. (ip=10*NFL+1 for the *from* volume

associated with flow path index NFL, ip=10*NFL+2 for the *to* volume associated with flow path index NFL, and ip=10*NCAV for the pool associated with cavity index NCAV.)

Masses of radioactive masses transported through flow paths are available as plot variables and/or control function arguments for paths that have been designated as *MACCS2* release paths. See the FL Package Users' Guide.

6 Control Function Variables

The RN package variables that may be used for control function arguments are listed and described below. Note that plot variables (some that are identical in definition to these control function arguments but different in format) are described in the previous section.

RN1-CPUC	Total time for the run routines of the RN1 package. (units = s)
RN1-CPUE	Total time for the edit routines of the RN1 package. (units = s)
RN1-CPUR	Total time for the restart routines of the RN1 package. (units = s)
RN1-CPUT	Sum of the run, edit, and restart times of the RN1 package. (units = s)
RN1-ATMG(NameCV)	Total mass of aerosol (radioactive plus nonradioactive) in the gas phase, for each control volume name *NameCV*. (units = kg)
RN1-ARMG(NameCV)	Total mass of radioactive aerosol in the gas phase, for each control volume name *NameCV*. (units = kg)

RN1-VTMG(NameCV)	Total mass of fission product vapor (radioactive plus nonradioactive) in the gas phase, for each control volume name *NameCV*. (units = kg)
RN1-VRMG(NameCV)	Total mass of radioactive fission product vapor in the gas phase, for each control volume name *NameCV*. (units = kg)
RN1-ATML(NameCV)	Total mass of aerosol (radioactive plus nonradioactive) in the liquid phase, for each control volume name *NameCV*. (units = kg)
RN1-ARML(NameCV)	Total mass of radioactive aerosol in the liquid phase, for each control volume name *NameCV*. (units = kg)
RN1-VTML(NameCV)	Total mass of fission product vapor (radioactive plus nonradioactive) in the liquid phase, for each control volume name *NameCV*. (units = kg)
RN1-VRML(NameCV)	Total mass of radioactive fission product vapor in the liquid phase, for each control volume name *NameCV*. (units = kg)
RN1-XMRLSE(NameCV,NameCLS,y)	Total mass of class *NameCLS* released from COR components or CAV debris in control volume *NameCV*. The parameter *y* specifies total mass as a compound (y= 'COM'), the radioactive mass only (y= 'RAD'), or the total mass as an element, including non-radioactive sources and/or structural releases (y= 'ELE'). (units = kg)
RN1-XMRLSET	Total non-radioactive plus radioactive mass released from COR components. (units = kg)
RN1-XMRLSER	Total radioactive mass released from COR components. (units = kg)

RN1-TOTMAS(NameCLS,y)

Sum of masses of class *NameCLS* in all locations, excluding that not yet released from fuel or debris. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD').

(units = kg)

RN1-TYCLAIR(NameCLS, NameTypeCVH,y)

Airborne mass of class *NameCLS* in all volumes of type NameTypeCVH, including aerosols and vapors. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD').

(units = kg)

RN1-AMG(NameCV,NameCLS,w,y)

Aerosol mass of section *w*, class name *NameCLS*, in the atmosphere of control volume name *NameCV* not including aerosols deposited on heat structures. The parameter *y* specifies total mass (y='TOT') or radioactive mass only (y='RAD').

(units = kg.)

RN1-VMG(NameCV,NameCLS,y)

Vapor mass of class name *NameCLS*, in the atmosphere of control volume name *NameCV* not including vapor deposited on heat structures. The parameter *y* specifies total mass (y='TOT') or radioactive mass only (y='RAD').

(units = kg)

RN1-AML(NameCV,NameCLS,y)

Aerosol mass of class name *NameCLS*, in the pool of control volume name *NameCV* not including aerosols deposited on heat structures. The parameter *y* specifies total mass (y='TOT') or radioactive mass only (y= 'RAD').

(units = kg)

RN1-VML(NameCV,NameCLS,y)

Vapor mass of class name *NameCLS*, in the pool of control volume name *NameCV* not including vapor deposited on heat structures. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD').

(units = kg)

RN1-ADEP(NameHS,s,NameCLS,y)

Aerosol mass of class *NameCLS*, deposited on side *s* (s='LHS' or s='RHS') of heat structure name *NameHS*. The parameter *y* specifies total mass (y='TOT') or radioactive mass only (y='RAD').

(units = kg)

RN1-DEPHS(NameHS,s,*NameCLS*,p)

Total radionuclide mass of class *NameCLS* deposited on side s (s='LHS' or s='RHS') of heat structure HS *NameHS* from deposition physics model p. The deposition models that are tracked are as follows:

p = 'DIFF', Diffusion deposition
p = 'THERM', Thermophoresis
p = 'GRAV', Gravitational settling
p = 'TURB', Turbulent deposition in straight sections
p = 'BEND', Deposition in pipe bends
p = 'VENT', Deposition in venturi transitions
p = 'CONT', Deposition in contraction transitions

(units = kg)

Example:

```
EXEC_PLOT 3
   1 RN1-DEPHS('40501',LHS,CS,GRAV) ! gravitational
   2 RN1-DEPHS('40501',LHS,CS,THERM)! thermophoresis
   3 RN1-DEPHS('40501',LHS,CS,DIFF) ! diffusive
```

RN1-VDEP(NameHS,s,NameCLS,y)

Vapor mass of class name *NameCLS*, deposited on side *s* (s='LHS' or s='RHS') of heat structure name *NameHS*. The parameter *y* specifies total mass (y='TOT') or radioactive mass only (y='RAD').

(units = kg)

RN1-ATMT

Total radioactive plus non-radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures.

(units = kg)

RN1-ATMR

Total radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures.

(units = kg)

RN1-VTMT	Total radioactive plus non-radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures. (units = kg)
RN1-VTMR	Total radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures. (units = kg)
RN1-TMT	Total radioactive and non-radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to *RN1-ATMT* plus *RN1-VTMT* variables. (units = kg)
RN1-TMR	Total radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to *RN1-ATMR* plus *RN1-VTMR* variables. (units = kg)
RN1-MDT(NameHS,s,y)	Total radioactive mass deposited on heat structure name *NameHS* on side *s* (LHS or RHS). The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-TMDTT	Total radioactive and non-radioactive mass deposited on all heat structures. (units = kg)
RN1-TMDTR	Total radioactive mass deposited on all heat structures. (units = kg)
RN1-DHTOT	Total decay heat calculated by the RN package for all locations of radionuclides. (units = W)
RN1-DHCOR	Total decay heat from the radionuclides in the core. (units = W)

RN1-DHCAV	Total decay heat from the radionuclides in the cavity. (units = W)
RN1-DHDEP	Total decay heat from the radionuclides deposited on the heat structures. (units = W)
RN1-DHATM	Total decay heat from airborne radionuclides. (units = W)
RN1-DHPOL	Total decay heat from radionuclides in the pool. (units = W)
RN1-AMGT(NameCV,NameCLS,y)	Aerosol mass of class name *NameCLS* in the atmosphere of control volume name *NameCV* (sum of sections), not including aerosols deposited on heat structures. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-CVCLT(NameCV,NameCLS,y)	Total mass of class name *NameCLS* as aerosol and vapor in control volume name *NameCV*. Includes mass in pool and atmosphere, but not the mass deposited on heat structures. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-TYCLT(NameCLS,NameTypeCVH,y)	Radioactive mass of class name *NameCLS* in all control volumes of type *NameTypeCVH*, including mass deposited on heat structures associated with the control volumes. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-CVTOT(NameCV,y)	Mass of radionuclides in control volume *NameCV*. Includes mass in pool and atmosphere, but not that deposited on heat structures. The parameter *y* specifies either

	total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-TYTOT(NameTypeCVH,y)	Sum of masses of radionuclides in all control volumes of type *NameTypeCVH*. Includes mass in pool and atmosphere, but not the mass deposited on heat structures. The parameter *y* specifies either total mass (y='TOT') or just the radioactive mass (y='RAD'). (units = kg)
RN1-MMDW(NameCV)	Mass median diameter of the wet aerosol distribution in the gas phase for each control volume name *NameCV*. (units = m)
RN1-GSDW(NameCV)	Geometric standard deviation of the wet aerosol distribution in the gas phase for each control volume name *NameCV*. (units = none)
RN1-MMDD(NameCV)	Mass median diameter of the dry aerosol distribution in the gas phase for each control volume name *NameCV*. (units = m)
RN1-GSDD(NameCV)	Geometric standard deviation of the dry aerosol distribution in the gas phase for each control volume name *NameCV*. (units = none)
RN1-PH(NameCV)	pH of pool in control volume name *NameCV*. (units = none)
RN1-IOP(NameCV,ccc)	Concentration of aqueous species *ccc* in control volume name *NameCV*. The *c* means that the pool species must be specified on record RN1_IOP before it can be output. This is done because there are many pool species (39), and the user does not necessarily want them all on the plot file, increasing the size of the plot output file. The species ccc can be one of the names SPCNAM in the table in the RN1_IOP record description.

	(units = kmole/m^3)
RN1-IOT(NameCV,IDEP,y)	Total mass of iodine pool surface deposition class number *IDEP* deposited on surfaces in control volume name *NameCV*. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD'). (units = kg)
RN1-IOD(NameHS,s,IDEP,y)	Mass of iodine pool surface deposition class number *IDEP* deposited on side *s* (LHS or RHS) of heat structure name *NameHS*. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD'). (units = kg)
RN1-CAT(NameCV,ICHEM,y)	Total mass of chemisorption surface deposition class number *ICHEM* deposited on surfaces in control volume name *NameCV*. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD'). (units = kg)
RN1-CAD(NameHS,s,ICHEM,y)	Mass of chemisorption surface class number *ICHEM* deposited on side *s* (LHS or RHS) of heat structure *NameHS*. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD'). (units = kg)
RN1-TMCAT	Total mass of chemisorbed species deposited on all heat structures. (units = kg)
RN1-TMCAR	Total radioactive mass of chemisorbed species deposited on all heat structures. (units = kg)
RN1-MCA(NameHS,s,y)	Total mass chemisorbed on side *s* (s='LHS' or s='RHS') of heat structure name *NameHS*. The parameter *y* specifies either total mass (y='TOT') or radioactive mass only (y='RAD'). (units = kg)

RN1-MMDC(NameCV,x)	Mass median diameter of component *x* in the aerosol distribution in the gas phase for each control volume name *NameCV*. (units = m)
RN1-GSDC(NameCV,x)	Geometric standard deviation of component *x* in the aerosol distribution in the gas phase for each control volume name *NameCV*. (units = none)
RN1-VCND(NameCV)	Water mass condensation rate on aerosol per unit volume in control volume name *NameCV* (calculated only for hygroscopic aerosol model). (units = kg/s)
RN2-CPUC	Total time for the run routines of the RN2 package. (units = s)
RN2-CPUE	Total time for the edit routines of the RN2 package. (units = s)
RN2-CPUR	Total time for the restart routines of the RN2 package. (units = s)
RN2-CPUT	Sum of the run, edit, and restart times of the RN2 package. (units = s)
RN2-AMFLT (FilterName,NameCLS)	Total aerosol mass of class name *NameCLS* on filter name *FilterName*. (units = kg.)
RN2-AMFLTS(FilterName)	Total aerosol mass on filter name *FilterName* (sum of RN2-AMFLT (FilterName,NameCLS)). (units = kg)
RN2-RAFLT(FilterName,NameCLS)	Radioactive aerosol mass of class name *NameCLS* on filter name *FilterName*. (units = kg.)
RN2-VMFLT(FilterName,NameCLS)	Total vapor mass of class *NameCLS* on filter *FilterName*.

	(units = kg)
RN2-RVFLT(FilterName,NameCLS)	Radioactive vapor mass of class name *NameCLS* on filter name *FilterName*. (units = kg.)
RN2-VMFLTS(FilterName)	Total vapor mass on filter name *FilterName* (sum of RN2-VMFLT(FilterName,NameCLS)). (units = kg)
RN2-FLT-QTOT(FilterName)	Total decay heat from all radionuclides deposited on filter name *FilterName*. (units = W)
RN2-FLT-QLOS(FilterName)	Heat loss from filter name *FilterName* (portion of RN2-FLT-QTOT that is assumed to be lost from the system). (units = W)
RN2-VFLT-TMP(FilterName)	Temperature of charcoal bed in vapor filter name *FilterName*. (units = K)
RN2-VFLT-RAD(FilterName)	Radiolytic desorption rate of fission product vapors from filter name *FilterName*. (units = kg/s)
RN2-VFLT-THR(FilterName)	Thermal desorption rate of fission product vapors from filter name *FilterName*. (units = kg/s)
RN2-VFLT-BUR(FilterName)	Rate of fission product vapor release from filter name *FilterName* due to charcoal combustion. (units = kg/s)
RN2-DFBUB-W(Key,Name,ISEC)	Instantaneous decontamination factor of aerosols in mass section *ISEC* from pool scrubbing associated with the path Name. *Key* takes one of the following values: Key = FROM for the *from* volume associated with the flow path Name, Key = TO for the *to* volume associated with the flow path Name, and *Key* = CAV for the pool associated with cavity Name.

RN2-DFBUB-A(Key,Name)	Instantaneous decontamination factor of total aerosol mass from pool scrubbing associated with the path Name. *Key* takes one of the following values: *Key* = FROM for the *from* volume associated with the flow path *Name*, *Key* = TO for the *to* volume associated with the flow path Name, and *Key* = CAV for the pool associated with the cavity Name.
RN2-DFBUB-V(Key,Name,NameCLS)	Instantaneous decontamination factor of vapor in class name *NameCLS* from pool scrubbing associated with the path Name. *Key* takes one of the following values: *Key* = FROM for the *from* volume associated with the flow path Name, *Key* = TO for the *to* volume associated with the flow path *Name*, and *Key* = CAV for the pool associated with the cavity Name.
RN2-DFBBT-W(Key,Name,ISEC)	Cumulative decontamination factor of aerosols in mass section *ISEC* from pool scrubbing associated with the path Name. *Key* takes one of the following values: *Key* = FROM for the *from* volume associated with the flow path Name, *Key* = TO for the *to* volume associated with the flow path *Name*, and *Key* = CAV for the pool associated with the cavity Name.
RN2-DFBBT-A(Key,Name)	Cumulative decontamination factor of total aerosol mass from pool scrubbing associated with the path *Name*. *Key* takes one of the following values: Key = FROM for the *from* volume associated with the flow path *Name*, *Key* = TO for the *to* volume associated with the flow path *Name*, and *Key* = CAV for the pool associated with the cavity Name.
RN2-DFBBT-V(Key,Name,NameCLS)	Cumulative decontamination factor of vapor in class name *NameCLS* from pool scrubbing associated with the path *Name*. *Key* takes one of the following values: *Key* = FROM for the *from* volume associated with the flow path *Name*, *Key* = TO for the *to* volume associated

with the flow path *Name*, and *Key* = CAV for the pool associated with the cavity Name.

Masses of radioactive masses transported through flow paths are available as plot variables and/or control function arguments for paths that have been designated as MACCS release paths. This capability is described in the FL Package Users' Guide.

7 Example Input

The following input is typical of that for a full plant calculation. Note that the CORSOR-Booth model is used instead of the default modified CORSOR-Booth model and the default class CsM is not used. A second aerosol component is used for better treatment of water aerosols, and the modeled size range for aerosols is extended. Initial radionuclide inventories in the core fuel are set for a three-ring, five-level core, with the bottom of active fuel at the sixth core level, and fractions of those inventories for five classes that have already been released to the fuel-cladding gap are initialized. Example input for flow-through areas and heat structure deposition surfaces is provided, and a filter having decontamination factor of 4.0 and able to remove up to 0.1 kg of aerosols independent of class is placed in a flow path.

```
DCH_INPUT
!*    17 - DCH Classes
DCH_CL 'XE' DEFAULT
DCH_CL 'CS' DEFAULT
DCH_CL 'BA' DEFAULT
DCH_CL 'I2' DEFAULT
DCH_CL 'TE' DEFAULT
DCH_CL 'RU' DEFAULT
DCH_CL 'MO' DEFAULT
DCH_CL 'CE' DEFAULT
DCH_CL 'LA' DEFAULT
DCH_CL 'UO2' DEFAULT
DCH_CL 'CD' DEFAULT
DCH_CL 'AG' DEFAULT
DCH_CL 'BO2' DEFAULT
DCH_CL 'H2O' DEFAULT
DCH_CL 'CON' DEFAULT
DCH_CL 'CSI' DEFAULT

RN1_INPUT ! RN1 package start record, activate RN package
!
! declare 2 aerosol components (one for water)
! and add 16th class for CsI
!       NUMSEC NUMCMP NUMCLS
RN1_DIM    5     2      16
!
! combine Cs and I to form CsI
RN1_CLS 'CLC1' 'CSI'  2 !N Donor class name  The number of moles
!  one mole class Cs per mole CsI
                         1      'CS'                  1.0
! one-half mole class I2 per mole CsI
                         2      'I2'                  0.5
! RN class CSI maps into VANESA group 25
```

```
RN1_CLVN 1 !NSTR  ICLSS  ICLSSV
             1     CSI    25

! VANESA group 25 maps back into RN class 16
RN1_VNCL   1 ! Number      Vanesa Group Number    RN Class Name
                 1                25                  'CSI'
RN1_FP00 CB_HBF  ! Use CORSOR-Booth model for high burn-up fuel
! initialize core radionuclide masses as fractions of total
! 40% rings 1 and 2, 20% ring 3
! 15% level 6, 25% level 7, 30% level 8, 20% level 9, 10%
! level10
RN1_FPN 16 ! Number Flag AxialNode Flag  Flag  RINP1 RINP2
! 4% of total mass in cell (10,1)
               1    1        10         DH     0.1   0.4
! 8% of total mass in cell (9,1)
               2    1         9         DH     0.2   0.4
! 12% of total mass in cell (8,1)
               3    1         9         DH     0.3   0.4
! 10% of total mass in cell (7,1)
               4    1         7         DH     0.25  0.4
! 6% of total mass in cell  (6,1)
               5    1         7         DH     0.15  0.4
! 4% of total mass in cell  (10,2)
               6    2         10        DH     0.1   0.4
! 8% of total mass in cell  (9,2)
               7    2         9         DH     0.2   0.4
! 12% of total mass in cell (8,2)
               8    2         8         DH     0.3   0.4
! 10% of total mass in cell (7,2)
               9    2         7         DH     0.25  0.4
! 6% of total mass in cell  (6,2)
              10    2         6         DH     0.15  0.4
! 2% of total mass in cell  (10,3)
              11    3        10         DH     0.1   0.2
! 4% of total mass in cell  (9,3)
              12    3         9         DH     0.2   0.2
! 6% of total mass in cell  (8,3)
              13    3         8         DH     0.3   0.2
! 5% of total mass in cell  (7,3)
              14    3         7         DH     0.25  0.2
! 3% of total mass in cell  (6,3)
              15    3         6         DH     0.15  0.2
! no initial cavity inventory
              16    0         CAV1      DH     0.   0.
! 100% total
!
! initialize gap fractions – constant throughout core

RN1_GAP 19 ! Number   IR   IA   Flag  Class RINP1 RINP2
! 3% of class Xe in gap
              1       1    10   CLASS  'XE'  0.03  1.0
! 5% of class Cs in gap
              2       1    10   CLASS  'CS'  0.05 1.128
! .0001% of class Ba in gap
              3       1    10   CLASS  'BA'  1.E-6 1.0
! 1.7% of class I in gap
              4       1    10   CLASS  'I2'  0.017 1.0
! .01% of class Te in gap
              5       1    10   CLASS  'TE'  0.0001 1.125
! same fractions as in cell (10,1)
              6       1     9   CELL   1 10  1.0
! same fractions as in cell (10,1)
              7       1     8   CELL   1 10  1.0
```

```
! same fractions as in cell (10,1)
             8       1     7   CELL   1 10  1.0
! same fractions as in cell (10,1)
             9       1     6   CELL   1 10  1.0
! same fractions as in cell (10,1)
            10      2    10   CELL   1 10  1.0
! same fractions as in cell (10,1)
            11      2     9   CELL   1 10  1.0
! same fractions as in cell (10,1)
            12      2     8   CELL   1 10  1.0
! same fractions as in cell (10,1)
            13      2     7   CELL   1 10  1.0
! same fractions as in cell (10,1)
            14      2     6   CELL   1 10  1.0
! same fractions as in cell (10,1)
            15      3    10   CELL   1 10  1.0
! same fractions as in cell (10,1)
            16      3     9   CELL   1 10  1.0
! same fractions as in cell (10,1)
            17      3     8   CELL   1 10  1.0
! same fractions as in cell (10,1)
            18      3     7   CELL   1 10  1.0
! same fractions as in cell (10,1)
            19      3     6   CELL   1 10  1.0
!
! change minimum and maximum aerosol sizes
!          DMIN    DMAX   RHONOM
RN1_ASP   5.E-7   5.E-4   1000.
!
RN1_ACOEF CALANDWR ! calculate the aerosol coefficients (don't
                   ! read)
!
! set water to aerosol component 2
RN1_CC ! NSTR  CLSNAM  AEROSOL COMPONENT NUMBER
         1       XE                1
         2       CS                1
         3       BA                1
         4       I2                1
         5       TE                1
         6       RU                1
         7       MO                1
         8       CE                1
         9       LA                1
         10      UO2               1
         11      CD                1
         12      AG                1
         13      BO2               1
         14      H2O               2
         15      CON               1
         16      CSI               1
!
! declare flowthrough areas at bottom of control volume CV301
RN1_SET  2  ! Number From volume  To volume  Elevation  Area
                1      'CV301'     'CV302'      12.0    75.0
                2      'CV301'     'CV303'      12.0    25.0
!
! override heat structure HS10501 orientation for RN deposition
RN1_DS 2 ! Number Name HS   Surface  Orientation
          1  'HS10501'  RHS  FLOOR  ! default orientation "WALL"
          2  'HS10501'  LHS CEILING ! default orientation "WALL"
!
RN2_INPUT
! place aerosol filter in flow path FL321
```

```
! with global DF of 4.0 and maximum loading of 0.1 kg
RN2_FLT   'FILTER1'  'FL321'   AEROSOL   VALUE   4.0   0.1
```

RN input for initial aerosol and vapor inventories and time-dependent sources is more appropriate for simulating experiments or just a portion of an accident in a plant. To initialize class aerosol masses in a control volume and to set up aerosol and vapor sources, the following input could be used:

```
RN1_INPUT
! declare tabular aerosol and vapor sources
!      NUMSEC NUMCMP NUMCLS
RN1_DIM   5     2      16
! initial CsI aerosol masses in control volume CV301
!        IVOL   ICLS    RFRAC
RN1_AG   CV301   CSI     1.0
!N     XMASS
 1      0.1
 2      0.1
 3      0.1
 4      0.1
 5      0.1
!
! CsI aerosol source defined by TF 'AEROSOL-SOURCE' with log-
! normal dist.
!                    IVOL IPHS ICLS RFRAC XM  ITAB  NAME
RN1_AS   AerSource1 CV301 2   CSI  1.0   1.0 TF 'AEROSOL-SOURCE'
! mass median diameter = 10.0 microns, standard deviation = 2.0
! AMMD GSD
RN1_AS01   LOGNORMAL   10.E-6   2.
!
!
! iodine vapor source defined by TF 'VAPOR-SOURCE'
RN1_VS 1  ! NSTR VOLUME  PHASE CLASS FRACTION MASS  TF/CF
             1   CV301    GAS   I2    1.0  1.0 TF 'VAPOR-SOURCE'
...
TF_INPUT
! aerosol TF linearly decreasing from 0.1 kg/s to zero at 100 s
!         TFNAME        TFSCAL   TFADCN
TF_ID 'AEROSOL-SOURCE'    1.0      0.0
TF_TAB   2 ! NTFPAR    X     Y
                1     0.0   0.1
                2   100.0   0.0
! vapor TF linearly constant at 0.05 kg/s
!         TFNAME        TFSCAL   TFADCN
TF_ID  'VAPOR-SOURCE'    1.0      0.0
TF_TAB   1 ! NTFPAR    X     Y
                1     0.0   0.05
```

8 Error Messages

The error messages from the MELGEN program are concerned with the input and should be self-explanatory. There are three error and informative messages from MELCOR, two concerning the aerosol calculations and one concerning the decay heat split.

The first message from MELCOR informs the user that the RN package had to cut the timestep while doing the aerosol time integration. The format is as follows:

```
RN1 PACKAGE TIME STEP CUT
CALLED BY AEROSOL RUNGE-KUTTA INTEGRATOR
IN CONTROL VOLUME nnnnn
```

where nnnnn is the control volume name.

It indicates that the numerical integration routine in MAEROS was unable to complete its integration through the timestep. If this error occurs frequently, relay this information to the MELCOR group so appropriate action can be taken.

The second error message is similar to the first, but occurs when the error in component mass conservation exceeds the specified tolerance given by C7000(2) for some component. The message is

```
RN1 PACKAGE TIME STEP CUT
EXCESSIVE ERROR IN AEROSOL CALCULATION
IN CONTROL VOLUME nnnnn
AEROSOL COMPONENT = xx RELATIVE ERROR = x.xxxxxEee
```

where nnnnn is the control volume name, xx is the component number.

And, if it appears frequently, the MELCOR development group should be notified.

The third error message notes a problem with the decay heat split in that all the decay heat is not used. The message is

```
PROBLEM WITH DECAY HEAT SPLIT AT TIME = X.XXXXEXX
TOTAL = X.XXXXEXX W
SUM OF SPLITS = X.XXXXEXX W
RELATIVE ERROR = X.XXXXEXX
```

Here RELATIVE ERROR is the difference between the total power from decay of radionuclides in control volumes or onto the surfaces (TOTAL) and the sum of the powers distributed to control volume contents and heat structure surfaces (SUM OF SPLITS). The occurrence of this message should be relayed to the MELCOR staff for resolution.

Appendix 1

MELCOR 2.0	PTFRead	Comments
RN1-CPUC	RN1-CPUC	Total time for the run routines of the RN1 package
RN1-CPUE	RN1-CPUE	Total time for the edit routines of the RN1 package
RN1-CPUR	RN1-CPUR	Total time for the restart routines of the RN1 package
RN1-CPUT	RN1-CPUT	Sum of the run, edit, and restart times of the RN1 package
RN1-ATMG(NameCV) NameCV – the control volume name	RN1-ATMG.cv cv – the control volume index	Total mass of the aerosol (radioactive plus nonradioactive) in the gas phase, for each control volume NameCV (index cv)
RN1-ARMG(NameCV) NameCV – the control volume name	RN1-ARMG.cv cv – the control volume index	Total mass of the radioactive aerosol in the gas phase for each control volume NameCV (index cv)
RN1-VTMG(NameCV) NameCV – the control volume name	RN1-VTMG.cv cv – the control volume index	Total mass of the fission product vapor (radioactive plus nonradioactive) in the gas phase for each control volume NameCV (index cv)
RN1-VRMG(NameCV) NameCV – the control volume name	RN1-VRMG.cv cv – the control volume index	Total mass of the radioactive fission product vapor in the gas phase for each control volume NameCV (index cv)
RN1-ATML(NameCV) NameCV – the control volume name	RN1-ATML.cv cv – the control volume index	Total mass of the aerosol (radioactive plus nonradioactive) in the liquid phase for each control volume NameCV (index cv)
RN1-ARML(NameCV) NameCV – the control volume name	RN1-ARML.cv cv – the control volume index	Total mass of the radioactive aerosol in the liquid phase for each control volume NameCV (index cv)
RN1-VTML(NameCV) NameCV – the control volume name	RN1-VTML.cv cv – the control volume index	Total mass of the fission product vapor (radioactive plus nonradioactive) in the liquid phase, for each control volume NameCV (index cv)
RN1-VRML(NameCV) NameCV – the control volume name	RN1-VRML.cv cv – the control volume index	Total mass of the radioactive fission product vapor in the liquid phase, for each control volume NameCV (index cv).
RN1-XMRLSE (NameCV,NameCLS,y) NameCV – the control volume name NameCLS – the class name; y = 'COM' y = 'RAD' y = 'ELE'	RN1-XMRLSE-x-y.cv cv – the control volume index x – the class index y = 1 y = 2 y = 3	Total mass of the class NameCLS (index x) released from COR components or CAV debris in the control volume NameCV (index cv). The parameter y specifies the total mass as a compound y='COM' (y =1), the radioactive mass only y='RAD' (y = 2), or the total mass as an element, including non-radioactive sources and/or structural releases y='ELE' (y = 3)
RN1-XMRLSET	RN1-XMRLSET	Total non-radioactive plus

MELCOR 2.0	PTFRead	Comments
		radioactive mass released from COR components
RN1-XMRLSER	RN1-XMRLSER	Total radioactive mass released from COR components
RN1-TOTMAS(NameCLS,y) NameCLS – the class name y = 'TOT' y = 'RAD'	RN1-TOTMAS-y.x x – the class index y = 1 y = 2	Sum of masses of the class NameCLS (index x) in all locations, excluding the mass not yet released from fuel or debris. The parameter y specifies either the total mass y='TOT' (y = 1) or just the radioactive mass y = 'RAD' (y=2)
RN1-ATMT	RN1-ATMT	Total radioactive plus non-radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures
RN1-ATMR	RN1-ATMR	Total radioactive aerosol mass in the atmosphere and pool regions, not including deposited aerosols on heat structures
RN1-VTMT	RN1-VTMT	Total radioactive plus non-radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures
RN1-VTMR	RN1-VTMR	Total radioactive fission product vapor mass in the atmosphere and pool regions, not including deposited vapors on heat structures.
RN1-TMT	RN1-TMT	Total radioactive and non-radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to RN1-ATMT plus RN1-VTMT variables
RN1-TMR	RN1-TMR	Total radioactive aerosol and fission product vapor masses in the atmosphere and pool regions. Equal to RN1-ATMR plus RN1-VTMR variables
RN1-MDT(NameHS,s,y) NameHS – the heat structure name s – the heat structure side s = LHS – left side s = RHS – right side y = 'TOT' y = 'RAD'	RN1-MDTT-n-y RN1-MDTR-n-y n – the heat structure index y – the heat structure side y = 1 for left side y = 2 for right side RN1-MDTT RN1-MDTR	Total radioactive mass deposited on the heat structure NameHS (index n) on side s = 'LHS' or = 'RHS' (y = 1 or 2). The parameter y specifies either the total mass y='TOT' (RN1-MDTT) or just the radioactive mass y='RAD' (RN1-MDTR)
RN1-TMDTT	RN1-TMDTT	Total radioactive and non-radioactive mass deposited on all heat structures
RN1-TMDTR	RN1-TMDTR	Total radioactive mass deposited on

MELCOR 2.0	PTFRead	Comments
		all heat structures.
RN1-DHTOT	RN1-DHTOT	Total decay heat calculated by the RN package for all locations of radionuclides
RN1-DHCOR	RN1-DHCOR	Total decay heat from the radionuclides in the core
RN1-DHCAV	RN1-DHCAV	Total decay heat from the radionuclides in the cavity
RN1-DHDEP	RN1-DHDEP	Total decay heat from the radionuclides deposited on the heat structures
RN1-DHATM	RN1-DHATM	Total decay heat from airborne radionuclides.
RN1-DHPOL	RN1-DHPOL	Total decay heat from radionuclides in the pool
RN1-TYCLT (NameCLS,NameTypeCVH,y) NameCLS – the class name; NameTypeCVH – the control volumes type name y = 'RAD' y = 'TOT'	RN1-TYCLT-x-2.ty x – the class index; ty – the control volumes type index RN1-TYCLT-x-2.ty	Radioactive mass of the class NameCLS (index x) in all control volumes of type NameTypeCVH (index ty), including the mass deposited on heat structures associated with the control volumes. The parameter y specifies either the total mass y='TOT' or just the radioactive mass y='RAD' (RN1-TYCLT-x-2.ty)
RN1-TYTOT (NameTypeCVH,y) NameTypeCVH – the control volumes type name y = 'RAD' y = 'TOT'	RN1-TYTOT-2.ty ty – the control volumes type index RN1-TYTOT-2.ty	Sum of the masses of the radionuclides in all control volumes of type NameTypeCVH (index ty). Includes the mass in pool and atmosphere, but not the deposited on heat structures. The parameter y specifies either the total mass y='TOT' or just the radioactive mass y='RAD' (RN1-TYTOT-2.ty)
RN1-MMDW(NameCV) NameCV – the control volume name	RN1-MMDW.cv cv – the control volume index	Mass median diameter of the wet aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN1-GSDW(NameCV) NameCV – the control volume name	RN1-GSDW.cv cv – the control volume index	Geometric standard deviation of the wet aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN1-MMDD(NameCV) NameCV – the control volume name	RN1-MMDD.cv cv – the control volume index	Mass median diameter of the dry aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN1-GSDD(NameCV) NameCV – the control volume name	RN1-GSDD.cv cv – the control volume index	Geometric standard deviation of the dry aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN1-PH(NameCV) NameCV – the control volume name	RN1-PH.nnn nnn – the control volume index	pH of pool in control volume name NameCV (index nnn)
RN1-IOP(NameCV,ccc)	RN1-IOP-cccccc.nnn	Concentration of aqueous species

MELCOR 2.0	PTFRead	Comments
NameCV – the control volume name ccc - aqueous species name	nnn – the control volume index cccccc – aqueous species name	ccc (cccccc) in the control volume NameCV (index nnn). The c means that the pool species must be specified on record RN1_IOP RNIOPyyy before it can be output. This is done because there are many pool species (39), and the user does not necessarily want them all on the plot file, increasing the size of the plot output file. The species ccc can be one of the names SPCNAM in the table in the RN1_IOP record description
RN1-TMCAT	RN1-TMCAT	Total mass of chemisorbed species deposited on all heat structures
RN1-TMCAR	RN1-TMCAR	Total radioactive mass of chemisorbed species deposited on all heat structures
RN1-MCA(NameHS,s,y) NameHS – the heat structure name s – the heat structure side s = LHS – left side s = RHS – right side y = 'TOT' y = 'RAD'	RN1-MCAT-n-y RN1-MCAR-n-y n – the heat structure index y – the heat structure side y = 1- left side y = 2 – right side RN1-MCAT-n-y RN1-MCAR-n-y	Total mass chemisorbed on side s='LHS' or s='RHS' (y = 1 or y = 2) of heat structure NameHS (index n). The parameter y specifies either the total mass y='TOT' (RN1-MCAT) or the radioactive mass only y='RAD' (RN1-MCAR)
RN1-MMDC(NameCV,x) NameCV – the control volume name x – the component number	RN1-MMDC-x.cv cv – the control volume index x – the component number	Mass median diameter of component x in the aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN1-GSDC(NameCV,x) NameCV – the control volume name x – the component number	RN1-GSDC-x.cv cv – the control volume index x – the component number.	Geometric standard deviation of component x in the aerosol distribution in the gas phase for each control volume NameCV (index cv)
RN2-CPUC	RN2-CPUC	Total time for the run routines of the RN2 package.
RN2-CPUE	RN2-CPUE	Total time for the edit routines of the RN2 package.
RN2-CPUR	RN2-CPUR	Total time for the restart routines of the RN2 package
RN2-CPUT	RN2-CPUT	Sum of the run, edit, and restart times of the RN2 package
RN2-AMFLT (FilterName,NameCLS) FilterName – the filter name NameCLS – the class name	RN2-AMFLT-x.f f – the filter index x – the class index	Total aerosol mass of the class NameCLS (index x) on the filter FilterName (index f)
RN2-RAFLT (FilterName,NameCLS) FilterName – the filter name	RN2-RAFLT-x.f f – the filter index	Radioactive aerosol mass of the class NameCLS (index x) on the filter FilterName (index f)

MELCOR 2.0	PTFRead	Comments
NameCLS – the class name	x – the class index	
RN2-AMFLTS(FilterName) FilterName – the filter name	RN2-AMFLT.f f – the filter index	Total aerosol mass on the filter FilterName (index f) (sum of RN2-AMFLT (FilterName, NameCLS))
RN2-VMFLT (FilterName,NameCLS) FilterName – the filter name NameCLS – the class name	RN2-VMFLT-x.f f – the filter index x – the class index	Total vapor mass of the class NameCLS (index x) on the filter FilterName (index f)
RN2-RVFLT (FilterName,NameCLS) FilterName – the filter name NameCLS – the class name	RN2-RVFLT-x.f f – the filter index x – the class index	Radioactive vapor mass of the class NameCLS (index x) on the filter FilterName (index f)
RN2-VMFLTS(FilterName) FilterName – the filter name	RN2-VMFLT.f f – the filter index	Total vapor mass on the filter FilterName (index f) (sum of RN2-VMFLT (FilterName, NameCLS))
RN2-FLT-QTOT(FilterName) FilterName – the filter name	RN2-FLT-QTOT.f f – the filter index	Total decay heat from all radionuclides deposited on the filter FilterName (index f)
RN2-FLT-QLOS(FilterName) FilterName – the filter name	RN2-FLT-QLOS.f f – the filter index	Heat loss from the filter FilterName (index f) (portion of RN2-FLT-QTOT that is assumed to be lost from the system)
RN2-VFLT-TMP(FilterName) FilterName – the filter name	RN2-VFLT-TMP.f f – the filter index	Temperature of the charcoal bed in the vapor filter FilterName (index f)
RN2-VFLT-RAD((FilterName) FilterName – the filter name	RN2-VFLT-RAD.f f – the filter index	Radiolytic desorption rate of fission product vapors from the filter FilterName (index f)
RN2-VFLT-THR(FilterName) FilterName – the filter name	RN2-VFLT-THR.f f – the filter index	Thermal desorption rate of fission product vapors from the filter FilterName (index f).
RN2-VFLT-BUR(FilterName) FilterName – the filter name	RN2-VFLT-BUR.f f – the filter index	Rate of fission product vapor release from the filter FilterName (index f) due to charcoal combustion
RN2-DFBUB-W (Key,Name,ISEC) Key = FROM Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value) ISEC – the section number	RN2-DFBUB-ip-sec.w ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV w – the section number	Instantaneous decontamination factor of aerosols in the mass section ISEC (w) from pool scrubbing associated with the path Name (ip)
RN2-DFBUB-A(Key,Name) Key = FROM Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value).	RN2-DFBUB-ip-a ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV	Instantaneous decontamination factor of the total aerosol mass from pool scrubbing associated with the path Name (ip)
RN2-DFBUB-V (Key,Name,NameCLS)	RN2-DFBUB-ip-cls.v	Instantaneous decontamination factor of vapor in the class name

MELCOR 2.0	PTFRead	Comments
Key = FROM Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value); NameCLS – the class name.	ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV v – the class number.	NameCLS (index v) from pool scrubbing associated with the path Name (ip)
RN2-DFBBT-W (Key,Name,ISEC) Key = FROM Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value) ISEC – the section number	RN2-DFBBT-ip-sec.w ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV w – the section number	Cumulative decontamination factor of aerosols in the mass section ISEC from pool scrubbing associated with the path Name. Key takes one of the following values: Key = FROM for the from volume associated with the flow path Name, Key = TO for the to volume associated with the flow path Name, and Key = CAV for the pool associated with the cavity Name
RN2-DFBBT-A(Key,Name) Key = FROM; Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value)	RN2-DFBBT-ip-a ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV	Cumulative decontamination factor of total aerosol mass from pool scrubbing associated with the path Name (ip)
RN2-DFBBT-V (Key,Name,NameCLS) Key = FROM Key = TO Key = CAV Name – the flow path or cavity name (depends on Key value) NameCLS – the class name	RN2-DFBBT-ip-cls.v ip=10*NFL+1 ip=10*NFL+2 ip=10*NCAV v – the class number	Cumulative decontamination factor of vapor in class name NameCLS (index v) from pool scrubbing associated with the path Name (ip)

Containment Sprays (SPR) Package

The MELCOR Containment Spray (SPR) package models the heat and mass transfer between spray water droplets and the containment building atmosphere.

This users' guide describes the input to the SPR package, including a brief description of the models employed, the input format, sample input, discussion of the output, sensitivity coefficients, plot variables, and control function arguments. Details on the models can be found in the SPR Package Reference Manual.

Contents

1 INTRODUCTION 4

2 USER INPUT 5

2.1 MELGEN Input 5

2.1.1 Spray Source Data 5

SPR_ID – Spray Source Data 6
SPR_DELETE – Ignore all records associated with SPR_ID 7
SPR_SRD – Spray Reservoir Data 7
SPR_DTFR – Spray Droplet Temperature and Flow Rate Data 8
SPR_DSD – Spray Droplet Size Distribution 10

2.1.2 Spray Junction Data 11

SPR_JUN – Spray Junction Data 11

2.1.3 Spray Sump Data 12

SPR_SUMP – Spray Sump Control Volume 12
SPR_CV – Control Volumes that Empty Sprays into Sump 12

2.1.4 Spray RN Data 13

SPR_RN2_CLS – Iodine Class 13
SPR_RN2_HPART – Spray Partition Coefficient 13

2.2 MELCOR Input 14

3 SENSITIVITY COEFFICIENTS 14

3.1 SPR Sensitivity Coefficients 14

3000 – Correlation Coefficients for Terminal Droplet Velocity and Reynolds Number 14
3001 – Correlation Coefficients for Mass Transfer 15
SPR_SC – SPR Sensitivity Coefficients Input Record 16

4 PLOT VARIABLES 17

5 CONTROL FUNCTION ARGUMENTS 18

6 EXAMPLE INPUT 18

6.1 Sample Problem Description 18

6.2 Sample Problem MELGEN User Input 18

6.3 Inventory-Limited Spray Sample Input 20

7 EXAMPLE OUTPUT 21

7.1 MELGEN Output 21
7.2 MELCOR Output 22

8 DIAGNOSTICS AND ERROR MESSAGES 24

APPENDIX 1 25

1 Introduction

The MELCOR Containment Spray (SPR) package models the heat and mass transfer between spray water droplets and the containment building atmosphere. The modeling in the SPR package is taken virtually intact from the HECTR 1.5 code. The model assumes, among other things, that spray droplets are spherical and isothermal, fall through containment at their terminal velocity, and that there is no horizontal velocity component. In a special application, the spray model is used to treat condensation from the Heat Structures (HS) package's film-tracking model that rains from inverted surfaces into the containment atmosphere.

An arbitrary number of spray sources may be placed in any containment control volume. The source of water (reservoir) for each spray may be identified with the pool of any Control Volume Hydrodynamics (CVH) package control volume, or it may be left unidentified. If a CVH pool is designated as the spray source reservoir, then the spray will operate only if the pool has sufficient water. The user provides input in the form of pool heights in the reservoir to determine whether there is sufficient water for spray operation. The user may set a flag to stop the calculation if the water level in any reservoir falls below the specified minimum requirement. Radionuclides dissolved in the reservoir water will not be transported along with the water to the spray source (i.e., it is assumed that there is a perfect filter for all radionuclides between the reservoir and the spray).

For each spray source, except for sources associated with rain from the HS film-tracking model, the user must specify an initial droplet temperature and flow rate, each of which may be controlled by a control function. The user may turn the sprays on and off with a separate control function for each spray source. A droplet size distribution may also be input for each spray source. In other words, the spray droplets for each source may be divided into a number of different size bins, with individual drops representing the average droplet size being tracked during their fall through the control volume; the total heat and mass transfer for the spray source is obtained by summing the heat and mass transfers calculated for all sizes.

For spray sources associated with rain from the HS film-tracking model, the user must define an out transfer process in MELCOR Transfer Process (TP) package that retrieves the rain from an associated in transfer process that is accessed by the HS film-tracking model. In this special application of the spray package model, the user should not specify control functions for the droplet temperature, flow rate, or a CVH pool as the source reservoir. Refer to Section 6.3 of the HS Package Users' Guide for an example of SPR package and TP package input for the special HS film-tracking rain application. Note that currently radionuclides associated with the condensate film in the HS package are not transferred with the rain to the SPR package; they stay with the remaining film on the structure.

For each droplet type, the following computation is made in each control volume in the spray train. Spray droplet heatup and cooldown in a steam environment are modeled using

a correlation for forced convection heat transfer coefficients. Similarly, evaporation and condensation are modeled using a correlation for mass transfer coefficients. The heat and mass transfer correlations have been formulated specifically for high temperature atmospheres, such as might be encountered during a hydrogen burn.

These coefficients are used to compute heat and mass transfer rates, which are integrated by a Runge-Kutta method over the fall height of the spray droplet to obtain the final droplet mass and temperature. By comparing the droplet mass and temperature at the bottom of the compartment to the inlet conditions, the heat transfer and mass transfer to a given droplet are computed. Total heat and mass transfer rates are calculated by multiplying the rates for one droplet by the total number of droplets of that size and summing over all droplet sizes. It is assumed that this total heat and mass transfer rate is constant over a given timestep, and it is also assumed that the containment atmosphere conditions do not change significantly during the fall time of the drop.

The user can describe how droplets falling from one control volume are to be carried over to lower control volumes. The user may designate that specified fractions of those droplets be transferred to one or more additional control volumes for continued heat and mass transfer to the atmosphere. (These droplets are currently treated independently of other spray trains involving those control volumes.) Alternatively, a separate control volume may be designated as the containment spray sump. Droplets reaching the bottom of a control volume and not carried over to other volumes will be placed in the pool of the sump control volume. Otherwise, droplets reaching the bottom of a control volume and not carried over to other volumes or placed in a separate sump are put into the pool of that control volume.

It should be noted that the SPR package, as all other MELCOR packages, does not model interactions between spray droplets and other structures. Thus, it is not possible to model either core sprays or steam generator auxiliary feed water sprays properly using the SPR package.

Warning: The SPR package is coupled to the MELCOR RadioNuclide (RN) package for the calculation of aerosol washout and atmosphere decontamination by the sprays. Current limitations of this interface require some restrictions on the input to the SPR package to avoid nonphysical results associated with multiple calculations in the same control volume. When the SPR and RN packages are both active, the user should limit the spray input so that only one spray train passes through each control volume and only a single drop size is used in this spray train.

2 User Input

2.1 MELGEN Input

2.1.1 Spray Source Data

The user must define the control volumes in which the spray sources are located. The user may optionally define the elevations of the spray sources if the sources are not at the top of the control volumes. The user may optionally input the names of the control functions

determining whether the spray sources are on or off. Also, for each source the user must define the droplet initial temperature, the spray flow rate, and the droplet size distribution.

SPR_ID – Spray Source Data

Required

This record allows the user to define a spray source name, the name of the control volume containing the spray source, and the elevation of the source in that control volume. The user can also define an optional control function determining whether the spray source is on or off.

(1) SPNAME

Character string defining a name for spray source.

(type = character*16, default = none, units = none)

(2) IVOL

Name of control volume containing the spray source.

(type = character*16, default = none, units = none)

(3) FALLHS

Elevation of the spray source in control volume IVOL source. If this number is not input, the top elevation of control volume IVOL will be used. The value of FALLHS must be between the bottom and top elevations of control volume IVOL.

(type = real, default = top elevation of control volume IVOL, units = m)

(4) ISPCON

Key for logical control function to be used in determining whether spray source is on or off.

(a) 0 or NO

Control function is not used and the spray source is assumed to always be on.

(b) 1 or CF

Control function is used in determining whether spray source is on or off.

(type = integer / character*2, default = 0, units = none)

if ISPCON = CF or 1, then

(5) NAMECF

Name of logical control function to be used in determining whether spray source is on or off. If the control function value is .TRUE., the spray source is on, if the value is .FALSE., the spray source is off.

(type = character*16, default = none, units = none)

Examples

```
SPR_ID SPRAY_1 CVH705  0.18E+02  1 CF298
SPR_ID SPRAY_2 CVH705  0.18E+02  NO
```

SPR_DELETE – Ignore all records associated with SPR_ID

Optional

This record is processed after all records in the input file are read. It will turn off all SPR records related to the SPNAME identifier.

(1) SPNAME

Character string defining a name for spray source.

(type = character*16, default = none, units = none)

Example

```
SPR_DELETE   SPRAY_1
```

SPR_SRD – Spray Reservoir Data

Optional

This record allows the user to define a control volume containing the water reservoir, a flag to stop the calculation if the reservoir becomes dry, the pool elevation below which the reservoir is declared to be dry, and the pool elevation that must be reached before spray operation can resume following dryout.

(1) IFDRY

Reservoir dryout option flag.

(a) 0

To inactivate spray upon reservoir dryout.

(b) 1

To stop calculation upon reservoir dryout.

(type = integer, default = 0, units = none)

(2) IVOLRS

Name of control volume containing reservoir (pool) for spray source. If NO is input, then the spray source is unidentified.

(type = character*16, default = NO, units = none)

(3) ELDRY

Reservoir pool elevation at dryout.

(type = real, default = CVBOT + 0.01*(CVTOP – CVBOT))

(4) ELWET

Reservoir pool elevation to resume spray after dryout.

(type = real, default = ELDRY + 0.04*(CVTOP – CVBOT))

Examples

```
SPR_SRD  0 NO   0.3402E+1 0.3403E+3
SPR_SRD  1 CV1  0.3402E+1 0.3403E+3
```

SPR_DTFR – Spray Droplet Temperature and Flow Rate Data

Required

This record allows the user to define the initial spray droplet temperature and flow rate for the spray source. These values may be defined either as constants or by control functions. The option TP for KEYDT is only used in conjunction with the HS package film-tracking rain model.

(1) KEYDT

Key for droplet temperature.

(a) -1 or TP

Out transfer process, associated with the in transfer process that the HS package uses to transfer rain from the film-tracking model to the SPR package, is used.

(b) 0 or CONST

Initial temperature of all droplets from this source is defined by constant.

(c) 1 or CF

Control function value is the initial temperature (K) of all droplets from this source.

(type = integer/character*5, default = 0, units = none)

optionally followed by

(2) IHSTP

Out transfer process name associated with the in transfer process that the HS package uses to transfer rain from the film-tracking model to the SPR package. If IHSTP is input, the specifications of temperature (TDROPO or ITMPCF) and flow (SPFLO or IFLOCF) will not be used. This field is required if KEYDT equals TP or -1.

(type = character*16, default = none, units = none)

or

(2) ITMPCF

Name of real-valued control function with an initial temperature (K) value of all droplets from this source. The value of the control function must be between 273.15 K and 647.245 K.
This field is required if KEYDT equals CF or 1.

(type = character*16, default = none, units = none)

or

(2) TDROPO

Initial temperature of all droplets from this source. The value of TDROPO must be between 273.15 K and 647.245 K.
This field is required if KEYDT equals CONST or 0.

(type = real, default = none, units = K)

If this record is not being used for the HS rain mode (KEYDT does not equal TP), then

(3) KEYFL

Key for flow rate data

(a) 0 or CONST

Total spray volumetric flow rate from this source is defined by constant.

(b) 1 or CF

Control function value is the total spray flow rate for this source.

(type = integer/character*5, default = 0, units = none)

optionally followed by

(4) IFLOCF

Name of a real-valued control function whose value is the total spray flow rate (m^3/s) for this source. The value of the control function must be greater than or equal to zero.
This field is required if KEYFL equals CF or 1.

(type = character*16, default = none, units = none)

or

(4) SPFLO

Total spray volumetric flow rate from this source. The value of SPFLO must be greater than or equal to zero.
This field is required if KEYFL equals CONST or 0.

(type = real, default = none, units = m^3/s)

Examples

```
SPR_DTFR -1    TPout205
SPR_DTFR CONST 0.215 CONST 0.45
SPR_DTFR CONST 0.215 CF    CF26
SPR_DTFR CF    CF25  CF    CF26
SPR_DTFR CF    CF25  CONST 0.45
```

SPR_DSD – Spray Droplet Size Distribution

Required

This record allows the user to define the initial spray droplet diameter and relative frequency for a droplet type in the spray source.

Note: When the RN package is active, radionuclide washout by sprays is automatically calculated. However, the radionuclide washout calculations should be used with caution when multiple spray droplet sizes and/or spray trains are present in the same control volume. For this reason, the user is strongly urged to avoid multiple droplet sizes and multiple spray trains in any control volumes where radionuclide washout calculations are to be performed.

(1) NSRCSZ

Number of droplet types.

(type = integer, default = none, units = none)

The following data are input as rows of the table with length NSRCSZ:

(1) NT

Droplet type index is used for sequencing.

(type = integer, default = none, units = none)

(2) DIAMO

Initial diameter for this droplet type from this source. The value of DIAMO must be greater than zero.

(type = real, default = none, units = m)

(3) DRFREQ

Relative frequency in source for this droplet type. The value of each frequency must be between 0.0 and 1.0, and the frequencies of all droplets from a source must sum to 1.

(type = real, default = none, units = none)

Example

```
SPR_DSD 4  !NT        DIAMO         DRFREQ
            1        0.9E-03        0.4
            2        0.8E-03        0.3
            3        0.7E-03        0.2
            4        0.6E-03        0.1
```

2.1.2 Spray Junction Data

Spray droplets reaching the bottom of a control volume may be carried over to other control volumes. The fraction of these droplets entering each subsequent control volume is specified by the user. If the sum of the specified fractions for a given from control volume is CAROVR, then CAROVR must be no greater than 1. If CAROVR is less than 1, and the from control volume is not in the list of control volumes emptying into the sump (see Section 2.1.3), then a fraction (1 – CAROVR) of the droplets is placed into the pool of the from control volume.

SPR_JUN – Spray Junction Data

Optional

(1) NSPJUN

The user-defined number of spray junctions.

(type = integer, default = 0, units = none)

The following data are input as rows of the table with length NSPJUN:

(1) NJ

Junction index is used for sequencing.

(type = integer, default = none, units = none)

(2) KCVFM

From control volume name for this junction.

(type = character*16, default = none, units = none)

(3) KCVTO

To control volume name for this junction.

(type = character*16, default = none, units = none)

(4) FRSPTI

Fraction of spray droplets reaching bottom of from volume that are to be transported into to volume. Must be between 0 and 1.

(type = real, default = none, units = none)

Example

```
SPR_JUN 5 !NJ  FromCVH    ToCVH    FRSPTI
           1   CVH705     CVH704    0.55
           2   CVH705     CVH702    0.45
           3   CVH702     CVH701    1.0
```

```
4   CVH704     CVH703    1.0
5   CVH703     CVH701    1.0
```

2.1.3 Spray Sump Data

The user may optionally define the control volume that contains the sump. The sump is a pool into which spray droplets will be deposited if the droplets reach the bottom of user-selected control volumes and are not carried over into other control volumes. The user may define a list of control volumes from which droplets will enter the sump. If the sum of the transmission factors for a volume in that list is CAROVR, then a fraction (1 – CAROVR) of the droplets reaching the bottom of the volume is placed into the sump. At present, no more than one sump may be defined.

SPR_SUMP – Spray Sump Control Volume

Optional

(1) MCVSUM

Name of the control volume containing the sump.

(type = character*16, default = none, units = none)

Example

```
SPR_SUMP CV100
```

SPR_CV – Control Volumes that Empty Sprays into Sump

Optional

This record is required if any spray droplets are to fall into the sump.

(1) NSPVSM

Number of control volumes.

(type = integer, default = 0, units = none)

The following data are input as rows of a table with length NSPVSM:

(1) NCV

Table record index.

(type = integer, default = none, units = none)

(2) ICVISM

Name of a control volume through which spray droplets may fall into sump.

(type = character*16, default = none, units = none)

Example

```
SPR_CV 2 !NCV  SumpCVH
          1    CV_1
          2    'Sump CVH'
```

2.1.4 Spray RN Data

The following records describe the washout of radionuclides by the SPR package. The user should refer to the RN Users' Guide for important information on the interactions between the SPR and RN packages.

SPR_RN2_CLS – Iodine Class

Optional

This record gives the class of iodine for vapor spray removal. All material in that class is assumed to be chemically similar to iodine and is subjected to the removal calculation.

(1) IICLS

Class of iodine for spray removal calculation.

(type = character*16, default = Halogens, units = none)

Example

```
SPR_RN2_CLS I2
```

SPR_RN2_HPART – Spray Partition Coefficient

Optional

This record specifies the partition coefficient used for the sprays. As many pairs as needed for multiple spray sources are input on this record.

(1) NUMSTR

Number of data strings in the spray partition coefficient table.

(type = integer, default = 0, units = none)

The following data are input as rows of a table with length NUMSTR:

(1) NP

Partition coefficient index.

(type = integer, default = none, units = none)

(2) ISPNUM

User-defined spray source name.

(type = character*16, default = none, units = none)

(3) HPART

Iodine partition coefficient for this spray source. The partition coefficient is defined as the ratio of the concentration of iodine in the liquid droplets to the concentration of iodine in the gas under equilibrium conditions. It is normally much greater than one.

(type = real, default = 1.0, units = none)

Table 2.1 Recommended Iodine Partition Coefficient Values

Spray Type	Conservative	Best Estimate
Sodium Hydroxide	2,000	5,000
Sodium Thiosulfate	—	100,000
Hydrazine	—	5,000
Boric Acid	500	2,500

Example

```
SPR_RN2_HPART 1 ! N  ISPNUM  HPART
                 1   SPR1    10.
```

2.2 MELCOR Input

There is at present no MELCOR input to the SPR package.

3 Sensitivity Coefficients

The sensitivity coefficient feature in MELCOR is a powerful feature that gives the user the ability to change selected parameters in the physics models that would otherwise require modification of the FORTRAN source code. Their use is described in Section 6 of the MELCOR Executive (EXEC) Users' Guide.

3.1 SPR Sensitivity Coefficients

The sensitivity coefficients for the containment sprays package are identified by numbers from 3000 through 3099.

3000 – Correlation Coefficients for Terminal Droplet Velocity and Reynolds Number

This correlation is used for velocity and Reynolds number calculations. Three correlations are considered to take into account different expressions for the drag coefficient based on the Reynolds number.

$$
\begin{aligned}
C_d &= C3000(1)\,\mathrm{Re}^{C3000(2)} && \text{for} && \mathrm{Re} < C3000(4) \\
&= C3000(5)\,\mathrm{Re}^{C3000(6)} && \text{for } C3000(4) < && \mathrm{Re} < C3000(8) \\
&= C3000(9) && \text{for } C3000(8) < && \mathrm{Re}
\end{aligned}
$$

The elements of sensitivity coefficient array 3000 are not independent. The following relationships should be observed:

$$C3000(3) = 2.0 + C3000(2)$$

$$C3000(7) = 2.0 + C3000(6)$$

$$C3000(4) = \left[C3000(1) \,/\, C3000(5)\right] ** \left\{1 \,/\, \left[C3000(6) - C3000(2)\right]\right\}$$

$$C3000(8) = \left[C3000(5) \,/\, C3000(9)\right] ** \left\{-1 \,/\, C3000(6)\right\}$$

(1) Multiplier for low Reynolds number
(default = 27.0, units = none, equiv = none)

(2) Exponent for low Reynolds number
(default = -0.84, units = none, equiv = none)

(3) 2 + C3000(2)
(default = 1.16, units = none, equiv = none)

(4) Breakpoint between low and medium Reynolds correlations
(default = 77.737154, units = none, equiv = none)

(5) Multiplier for medium Reynolds number
(default = 0.271, units = none, equiv = none)

(6) Exponent for medium Reynolds number
(default = 0.217, units = none, equiv = none)

(7) 2 + C3000(6)
(default = 2.217, units = none, equiv = none)

(8) Breakpoint between medium and high Reynolds correlations
(default = 10006.443, units = none, equiv = none)

(9) Drag coefficient at high Reynolds numbers
(default = 2.00, units = none, equiv = none)

3001 – Correlation Coefficients for Mass Transfer

This correlation is used to calculate the mass derivative. It uses the mass transfer correlation described in the SPR Package Reference Manual.

$$F(\mathrm{Re}, Sc) = C3001(3) + C3001(4)\mathrm{Re}^{C3001(5)}\, Sc^{C3001(6)}$$

C3001(1) is the leading multiplier in the correlation and must be negative, and C3001(2) should not be changed.

(1) Leading multiplier for derivative of mass with respect to time equation
(default = -2.0, units = none, equiv = none)

(2) Additive factor in log expression
(default = 1.0, units = none, equiv = none)

(3) Additive factor in Reynolds-Schmidt number combination
(default = 1.0, units = none, equiv = none)

(4) Multiplicative factor in Reynolds-Schmidt number combination
(default = 0.25, units = none, equiv = none)

(5) Exponent for Reynolds number
(default = 0.5, units = none, equiv = none)

(6) Exponent for Schmidt number
(default = 1./3., units = none, equiv = none)

SPR_SC – SPR Sensitivity Coefficients Input Record

Optional

This record allows the user to change the default SPR sensitivity coefficient values. This record is not required.

(1) NUMSTR
Number of data strings.
(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR
Data string index.
(type = integer, default = none, units = none)

(2) NNNN
Unique four-digit identifier of the SPR sensitivity coefficient array.
(type = integer, default = none, units = none)

(3) VALUE

New value of the sensitivity coefficient. Values must be real-valued only—no integer values are allowed.

(type = real, default = none, units = specific sensitivity coefficient dependent)

(4) NA

Sensitivity coefficient index.

(type = integer, default = none, units = none)

Example

```
SPR_SC   2  ! N   SCnumber    Value        Index
              1   3000         1.5           9
              2   3001         0.7           5
```

4 Plot Variables

The SPR package variables that may be used for plot variables are described below. Note that control function arguments (some are identical in definition to these plot variables but different in format) are described in the following section.

SPR-TP.n	Temperature of spray droplets from source index n. (units = K)
SPR-FL.n	Flow rate of spray droplets from source index n. (units = m^3/s)
SPR-HTTRAN.j	Rate of heat transfer from sprays to steam in spray control volume index j. (units = W)
SPR-MSTRAN.j	Rate of mass transfer from sprays to steam in spray control volume index j. (units = kg/s)]
SPR-SUMPHT	Rate of heat transfer from sprays to pool in sump control volume. (units = W)
SPR-SUMPMS	Rate of mass transfer from sprays to pool in sump control volume. (units = kg/s)

5 Control Function Arguments

The SPR package variables that may be used for plot variables and control function arguments are described below. Note that plot variables (some are identical in definition to these control function arguments but are different in format) are described in the previous section.

SPR-TP(NameSpr)	Temperature of spray droplets from source name NameSpr. (units = K)
SPR-FL(NameSpr)	Flow rate of spray droplets from source name NameSpr. (units = m^3/s)
SPR-HTTRAN (NameCVH)	Rate of heat transfer from sprays to steam in control volume name NameCVH. (units = W)
SPR-MSTRAN(NameCVH)	Rate of mass transfer from sprays to steam in volume name NameCVH. (units = kg/s)
SPR-SUMPHT	Rate of heat transfer from sprays to pool in sump control volume. (units = W)
SPR-SUMPMS	Rate of mass transfer from sprays to pool in sump control volume. (units = kg/s)

6 Example Input

6.1 Sample Problem Description

Containment is nodalized into five control volumes, CV100, CV110, CV120, CV130, and CV140. Two spray sources with different characteristics are defined. Source 2, MAINSPRAY, in volume CV100, has three droplet sizes. Source 6, BACKUPSPRAY, in volume CV110, has two droplet sizes. Seventy percent of the droplets leaving volume CV100 fall into volume CV120, and the rest fall into volume CV110. All droplets leaving volumes CV110 and CV120 fall into volume CV130. Volume CV140 is designated as the sump, and all droplets leaving volume CV130 enter the sump volume.

6.2 Sample Problem MELGEN User Input

```
!
! MELGEN INPUT
```

```
!
! ************************************************
! * SRC. 2    *
! * MAINSPRAY *
! * ______* * *
! * CV100 *
! * *
! ************************************************
! * SRC. 6     * *
! * BACKUPSPRAY* *
! *_______ * * * *
! * * *
! * CV110 * CV120 *
! * * *
! * * *
! ************************************************
! * *
! * *
! * CV130 *
! * *
! ************************************************
! * *
! * CV140 *
! * SUMP *
! * *
! ************************************************
! CONTAINMENT SPRAY INPUT
SPR_INPUT
!
! TWO SPRAY SOURCES WITH SEVERAL DROP SIZES, JUNCTIONS
! WITH DROPLETS FALLING INTO A SUMP
!
! SUMP INFO
!
! SUMP CONTROL VOLUME
SPR_SUMP   CV140
! CONTROL VOLUME OVER SUMP
SPR_CV   1   !NCV  SumpCVH
               1    CV130
!
! SPRAY JUNCTION DATA
!
! FROM VOL TO VOL TRAN FAC
SPR_JUN 5 !NJ  FromCVH    ToCVH    FRSPTI
           1    CV100     CV110     0.3
           2    CV100     CV120     0.7
           3    CV110     CV130     1.0
           4    CV120     CV130     1.0
!
! SPRAY SOURCE DATA
!
! SOURCE 2
! NAME VOL ELEV ON/OFF CF
SPR_ID   MAINSPRAY   CV100   10.0   NO
!
! TEMP FLOW TEMPERATURE CF FLOW CF
SPR_DTFR   CONST   300.   CONST   2.5
!
! THREE DROPLET SIZES
SPR_DSD 3  !NT       DIAMO      DRFREQ
            1        1.0E-3      0.6
            2        .5E-3       0.3
            3        .25E-3      0.1
```

```
!
! SOURCE 6
! NAME VOL ELEV ON/OFF CF
SPR_ID   BACKUPSPRAY   CV110   10.0   NO
!
! TEMP FLOW TEMPERATURE CF FLOW CF
SPR_DTFR   CONST   320.   CONST   1.3
!
!
! TWO DROPLET SIZES
! DIAM REL. FREQ.
SPR_DSD 2  !NT       DIAMO      DRFREQ
            1        0.75E-3     0.75
            2        0.66E-3     0.25
```

6.3 Inventory-Limited Spray Sample Input

It is often necessary to simulate situations where the inventory of spray source water is limited. For example, if the spray source water is taken from a tank, the sprays must be shut off when the tank runs dry. The following control functions simulate this situation. It is assumed that control function CF100 has already been formulated to represent the demanded spray volumetric flow rate. The remaining input integrates the spray flow rate and shuts off the sprays when the total spray flow reaches 1000.0 m^3.

```
!*
! CF CF100 (NOT SHOWN) IS DEMANDED SPRAY VOLUMETRIC FLOW RATE
!
! THIS CONTROL FUNCTION INTEGRATES THE DEMANDED FLOW RATE
!
CF_INPUT
CF_ID 'TOTAL INTEG' INTEG
CF_SAI   1.0   0.0   0.0
CF_ARG    2 ! NARG CHARG           ARSCAL   ARADCN
               1 CF-VALU(CF100)      1.0     0.0
               2 EXEC-TIME           1.0     0.0
!
! THIS CONTROL FUNCTION GIVES (1000 M**3 – INTEGRAL FLOW), SO
! IT IS NEGATIVE IF MORE THAN 1000 M**3 HAS BEEN DEMANDED. THEN
! IT TAKES THE 'SIGN' FUNCTION OF THE DIFFERENCE, GIVING A VALUE
! +1 IF INTEGRAL IS LESS THAN 1000 M**3, AND –1 IF GREATER. THE
! VALUE IS THEN LIMITED TO BE BETWEEN 0.0 AND 1.0. THUS, THE
! RESULT OF THIS CONTROL FUNCTION IS:
! 0.0 IF INTEGRAL IS GREATER THAN 1000 M**3 (FLOW IMPOSSIBLE)
! 1.0 IF INTEGRAL IS LESS THAN 1000 M**3 (FLOW IS POSSIBLE)
!
CF_ID 'MULTIPLIER' SIGNI
CF_SAI   1.0   0.0   1.0
CF_ULB   BOTH   0.0   1.0
CF_ARG   1 ! NARG CHARG                 ARSCAL      ARADCN
              1 CF-VALU('TOTAL INTEG')   -1.0       1000.0
! NOW MULTIPLY DEMANDED SOURCE BY FLOW POSSIBLE MULTIPLIER
!(ABOVE)
CF_ID 'SPSOURCE' MULTIPLY
CF_SAI   1.0   0.0
CF_ARG    2 ! NARG CHARG              ARSCAL     ARADCN
               1 CF-VALU('CF100')      1.0         0.0
               2 CF-VALU('MULTIPLIER') 1.0         0.0
!
! THIS CONTROL FUNCTION (SPSOURCE) MUST BE REFERENCED ON SPRAY
! PACKAGE. RECORD SPR_DTFR AND REPRESENTS THE ACTUAL VOLUMETRIC
! FLOW RATE TO BE SENT TO THE SPRAYS
```

7 Example Output

In the following output listings, the term volume weighted frequency is defined as the sum over all droplet sizes of the droplet frequency multiplied by the cube of the droplet diameter. The sign convention is positive for heat and mass transfer from the spray droplets to the control volume.

7.1 MELGEN Output

```
      ***** CON SPRAY SETUP EDIT *****

  TOTAL NUMBER OF CON SPRAY SOURCES   =     2
  TOTAL NUMBER OF CON SPRAY VOLUMES   =     4
  TOTAL NUMBER OF CON SPRAY JUNCTIONS =     3

 EDIT OF CON SPRAY SOURCE SPR1
    SPRAY SOURCE CONTROL VOLUME =     CV705
    SPRAY SOURCE ELEVATION =     2.10000E+01
    SPRAY SOURCE CONTROL FUNCTION = CF298
    DROPLET TEMPERATURE =               3.13150E+02
    DROPLET TEMPERATURE CONTROL FUNCTION = CF369
    DROPLET FLOW RATE =                 0.00000E+00
    DROPLET FLOW RATE CONTROL FUNCTION =   CF368
    VOLUME WEIGHTED FREQUENCY =         7.29000E-10
    DATA FOR THE   1 DROPLET GROUPS IN THIS SOURCE
            **** DROPLET GROUP  1 ****
      INITIAL DROPLET DIAMETER = 9.00000E-04
      DROPLET FREQUENCY =           1.00000E+00
      DROPLET FLOW RATE =           0.00000E+00
      INITIAL DROPLET MASS =        3.78665E-07

 EDIT OF JUNCTION DATA FOR THE     3 JUNCTIONS IN THE FLOW PATH FOR THIS SOURCE
 JUNCTION   1 FROM SPRAY VOL CV705             TO SPRAY VOL CV704             TRAN FAC
1.00000E+00
 JUNCTION   2 FROM SPRAY VOL CV704             TO SPRAY VOL CV703             TRAN FAC
3.30000E-01
 JUNCTION   3 FROM SPRAY VOL CV704             TO SPRAY VOL CV702             TRAN FAC
6.70000E-01

EDIT OF CON SPRAY SOURCE SPR7
    SPRAY SOURCE CONTROL VOLUME =     CV704
    SPRAY SOURCE ELEVATION =     1.84000E+01
    SPRAY SOURCE CONTROL FUNCTION = CF296
    DROPLET TEMPERATURE =               3.13150E+02
    DROPLET TEMPERATURE CONTROL FUNCTION = ----
    DROPLET FLOW RATE =                 5.00000E-01
    DROPLET FLOW RATE CONTROL FUNCTION =   ----
    VOLUME WEIGHTED FREQUENCY =         5.35400E-10
    DATA FOR THE   4 DROPLET GROUPS IN THIS SOURCE
            **** DROPLET GROUP  1 ****
      INITIAL DROPLET DIAMETER = 9.00000E-04
      DROPLET FREQUENCY =           4.00000E-01
      DROPLET FLOW RATE =           7.13433E+08
      INITIAL DROPLET MASS =        3.81704E-07
            **** DROPLET GROUP  2 ****
      INITIAL DROPLET DIAMETER = 8.00000E-04
      DROPLET FREQUENCY =           3.00000E-01
      DROPLET FLOW RATE =           5.35075E+08
```

```
      INITIAL DROPLET MASS =        2.68083E-07
            **** DROPLET GROUP  3 ****
      INITIAL DROPLET DIAMETER = 7.00000E-04
      DROPLET FREQUENCY =           2.00000E-01
      DROPLET FLOW RATE =           3.56716E+08
      INITIAL DROPLET MASS =        1.79594E-07
            **** DROPLET GROUP  4 ****
      INITIAL DROPLET DIAMETER = 6.00000E-04
      DROPLET FREQUENCY =           1.00000E-01
      DROPLET FLOW RATE =           1.78358E+08
      INITIAL DROPLET MASS =        1.13097E-07

 EDIT OF JUNCTION DATA FOR THE     2 JUNCTIONS IN THE FLOW PATH       FOR THIS SOURCE
 JUNCTION   1 FROM SPRAY VOL CV704             TO SPRAY VOL CV703             TRAN FAC
3.30000E-01
 JUNCTION   2 FROM SPRAY VOL CV704             TO SPRAY VOL CV702             TRAN FAC
6.70000E-01

EDIT OF CON SPRAY JUNCTION NUMBER    1
    'FROM' CONTROL VOLUME = CV705
    'TO' CONTROL VOLUME = CV704
    TRANSMISSION FACTOR =     1.00000E+00

EDIT OF CON SPRAY JUNCTION NUMBER    2
    'FROM' CONTROL VOLUME = CV704
    'TO' CONTROL VOLUME = CV703
    TRANSMISSION FACTOR =     3.30000E-01

EDIT OF CON SPRAY JUNCTION NUMBER    3
    'FROM' CONTROL VOLUME = CV704
    'TO' CONTROL VOLUME = CV702
    TRANSMISSION FACTOR =     6.70000E-01

EDIT OF SPRAY VOLUME AND CONTROL VOLUME TABLE
SPRAY VOLUME  CONTROL VOLUME  CARRY-OVER FRACTION    SUMP INDEX

      1        CV702               0.000E+00             0
      2        CV703               0.000E+00             1
      3        CV704               1.000E+00             0
      4        CV705               1.000E+00             0


 EDIT OF CONTROL VOLUMES CONNECTED TO SUMP CONTROL VOLUME =  CV701

   CV703
```

7.2 MELCOR Output

```
 ****************************************
 *   CONTAINMENT SPRAY PACKAGE EDIT     *
 ****************************************

                         CONTAINMENT SPRAY SOURCES

 SRC              C.V.             CONTROL          TEMP             DROPLET     FLOW
                                    C.F.            C.F.             TEMP (K)    C.F.

 SPR1             CV705            CF298            CF369            3.131E+02   CF368
 SPR7             CV704            CF296            ----             3.131E+02   ----
```

```
FL-RATE    HS RAIN        NO. DROP  STAT
(M**3/S)   T.P.            SIZES

8.0000E-01 ----               1      ON
5.0000E-01 ----               4      ON

                    DROPLET DISTRIBUTIONS FOR SOURCE SPR1

  GROUP    DIAMETER      REL. FREQ.     FLOW RATE        DROPLET MASS
   NO.        M                         DROPS/SEC           KG

    1     9.0000E-04     1.0000E+00     2.0959E+09     3.7874E-07

                    DROPLET DISTRIBUTIONS FOR SOURCE SPR7

  GROUP    DIAMETER      REL. FREQ.     FLOW RATE        DROPLET MASS
   NO.        M                         DROPS/SEC           KG

    1     9.0000E-04     4.0000E-01     7.1343E+08    3.7874E-07
    2     8.0000E-04     3.0000E-01     5.3507E+08    2.6600E-07
    3     7.0000E-04     2.0000E-01     3.5672E+08    1.7820E-07
    4     6.0000E-04     1.0000E-01     1.7836E+08    1.1222E-07

                    SPRAY JUNCTION DATA FOR SOURCE SPR1

 JUNCTION NO.   FROM VOLUME    TO VOLUME     TRANSMISSION FACTOR

     1           CV705          CV704           1.00000E+00
     2           CV704          CV703           3.30000E-01
     3           CV704          CV702           6.70000E-01

                    SPRAY JUNCTION DATA FOR SOURCE SPR7

JUNCTION NO.   FROM VOLUME    TO VOLUME     TRANSMISSION FACTOR

    2           CV704          CV703           3.30000E-01
    3           CV704          CV702           6.70000E-01

                    SPRAY HEAT AND MASS TRANSFER DATA


  CONTROL VOL       HEAT TRAN RATE   MASS TRAN RATE
                          W             KG/S

  CV702             0.00000E+00      0.00000E+00
  CV703             0.00000E+00      0.00000E+00
  CV704            -4.27373E+07     -1.55180E+01
  CV705            -6.67427E+07     -2.55755E+01

     SPRAY DROPLETS LEAVING THESE CONTROL VOLUMES WILL ENTER SUMP VOLUME CV701

   CV703
                    SPRAY SUMP HEAT AND MASS TRANSFER DATA

  CONTROL VOL                HEAT TRAN RATE   MASS TRAN RATE
                                   W             KG/S

 CV701                       0.00000E+00      0.00000E+00

 END OF EDIT FOR SPR
```

8 Diagnostics and Error Messages

A diagnostic message is printed whenever the differential equation solver is having difficulty integrating the droplet mass equation as the droplet falls through a control volume. This can occur when the problem is determined to be unsolvable.

A diagnostic message is printed if a droplet enters a control volume in which the saturation temperature corresponding to total pressure is lower than the temperature of the spray droplet. In that case, the heat and mass transfer rates are not calculated, though calculation of the fall of the droplet will continue.

A diagnostic message is printed if the code is not able to determine the equilibrium temperature of the spray droplet in the steam environment.

Appendix 1

Melcor 2.0	PTFRead	Comments
SPR-TP(NameSpr) NameSpr – the source name.	SPR-TP.n n – the source index.	Temperature of spray droplets from source name NameSpr (index n).
SPR-FL(NameSpr) NameSpr – the source name.	SPR-FL.n n – the source index.	Flow rate of spray droplets from source name NameSpr (index n).
SPR-HTTRAN (NameCVH) NameCVH – the spray control volume name.	SPR-HTTRAN.j j – the spray control volume index.	Rate of heat transfer from sprays to steam in control volume name NameCVH (index j).
SPR-MSTRAN(NameCVH) NameCVH – the spray control volume name.	SPR-MSTRAN.j j – the spray control volume index.	Rate of mass transfer from sprays to steam in volume name NameCVH (index j).
SPR-SUMPHT	SPR-SUMPHT	Rate of heat transfer from sprays to pool in sump control volume.
SPR-SUMPMS	SPR-SUMPMS	Rate of mass transfer from sprays to pool in sump control volume.

Tabular Function (TF) Package
Users' Guide

The Tabular Function (TF) package allows the user to input one-dimensional tables, specify boundary conditions at the end points of the tables, and use the tables from any portion of the code.

This users' guide gives a description of the TF package input, with some examples.

Contents

1 INTRODUCTION 3

2 MELGEN USER INPUT 4

TF_ID – Tabular Function Parameter Record 4
TF_BND – Boundary Condition Record 4
TF_TAB – Table of Data Pairs 5

3 MELCOR USER INPUT 6

4 EXAMPLE INPUT 6

1 Introduction

The Tabular Function (TF) package allows the user to define one-dimensional tables of (x,y) data pairs for arbitrary independent and dependent variables and to specify extrapolation conditions at the end points of these tables. These tables may then be accessed from other MELCOR packages as specified by the user via input to those packages. Common uses of tabular functions in MELCOR include specifying decay heat power, control volume mass/energy sources, and flow path velocities as functions of time; defining material properties as functions of temperature; and specifying heat structure boundary conditions (heat transfer coefficient, heat flux, or surface temperature) as functions of time or temperature. Also, user-defined control functions may access tabular functions to specify functional relationships between arbitrary pairs of variables.

Tabular function tables are defined by entering data pairs for both independent and dependent variables. The value of the tabular function (i.e., the y-value for a given x-value) is generated by linear interpolation between the data pairs. The interpolated value is then multiplied by a factor specified by the user and added to another user-defined constant. One is the minimum number of data pairs defined by the user. (For very large tables, the External Data File (EDF) package can be used; see the EDF Package Users' Guide.)

In most situations, input data pairs are sorted so that the independent variables are monotonically increasing. However, step functions can be included in a table by entering two or more values of the independent variable which are equal. In this case, the values are not automatically sorted, and entering a decreasing sequence of the independent variable is treated as a fatal error. If a step function is entered and a value is requested which is exactly equal to the step position, the last value entered is returned.

Three options for extrapolation conditions at the upper and lower boundaries of the table are available:

(1) The domain of the function can be extended indefinitely with the function value set equal to the value at the boundary and held constant (the default).

(2) The domain of the function can be extended indefinitely with the two data pairs at the boundary used to linearly extrapolate the table.

(3) The domain of the function can be limited to that spanned by the data pairs' independent variables. If the independent variable is outside the domain of definition, then a fatal error flag is set, and the value is set equal to the boundary value.

Different options may be applied at each end of a table.

2 MELGEN User Input

The user must input two or more records to define a tabular function. The records and their fields are required unless otherwise stated. The records are described in this section.

TF_ID – Tabular Function Parameter Record

Required

This record defines the tabular function name, additive, and multiplicative constant. The value of the tabular function is equal to

$$TFVALU = TFSCAL * f(x) + TFADCN$$

where f(x) is the interpolated value of the data pairs. If TFADCN is not defined on the record, a default value of zero is used.

(1) TFNAME

User-defined tabular function name.

(type = character*16, default = none)

(2) TFSCAL

Multiplicative scale factor.

(type = real, default = none, units = none)

(3) TFADCN

Additive constant. This field is optional.

(type = real, default = 0.0, units = none)

Example

```
TF_ID  'Bills function'  1.
```

TF_BND – Boundary Condition Record

Optional

The tabular function may be extended beyond its endpoints by extending the function indefinitely and holding the value constant at the endpoint, linearly extrapolating the value from the endpoint and the value nearest the endpoint, or returning an error if a value outside the domain of definition is requested. The upper and lower boundary switches may be individually set as follows:

(1) NTFBDL

Lower boundary condition switch.

(a) 0 or CONST-BND
Extend the domain indefinitely and hold constant at the boundary value (the default).

(b) 1 or EXTRAPOL-BND
Extend the domain indefinitely and linearly extrapolate the two data pairs at the boundary.

(c) 2 or ERR-BND
Running off the domain of the tabular function is prohibited.

(type = integer / character*12, default = 0, units = none)

(2) NTFBDU

Upper boundary condition switch.

(a) 0 or CONST-BND
Extend the domain indefinitely and hold constant at the boundary value (the default).

(b) 1 or EXTRAPOL-BND
Extend the domain indefinitely and linearly extrapolate the two data pairs at the boundary.

(c) 2 or ERR-BND
Running off the domain of the tabular function is prohibited.

(type = integer / character*12, default = 0, units = none)

NOTE: If this record is present, both values must be entered. If it is absent, both switches will be taken as default.

Example

```
TF_BND   CONST-BND   EXTRAPOL-BND
```

TF_TAB – Table of Data Pairs

Required

The tabular function is defined by data pairs. The number of tabular function pairs entered in this record must be equal to the number of pairs entered for this tabular function. If they do not agree, then an error occurs. The first number in a pair is the independent variable, the second number is the dependent variable. Each pair must be positioned on a different string (as a table).

(1) NTFPAR

Number of tabular function data pairs.

(type = integer, default = none, units = none)

The following data are entered as a table with length NTFPAR:

(1) NPAR

Data pair index is used for sequencing or replacing.

(type = integer, default = none, units = none)

(2) X

The first element (independent variable) in a data pair.

(type = real, default = none, dimension = none, units = as appropriate for the tabular function data)

(3) Y

The second element (dependent variable) in a data pair.

(type = real, default = none, units = as appropriate for the tabular function data)

Example

```
TF_TAB  4  !NPAR   X      Y
            1     0.0    1.0
            2     5.0    100.5
            3     100.8  55.
            4     5000.  90.993
```

3 MELCOR User Input

At present the TF package data cannot be altered by the user in MELCOR .

4 Example Input

The following input records define four tabular functions. The first has four data pairs, and the second has only one (constant function). The last two are step functions with a control function that could be used with a valve junction to open and close the valve. For more information on control functions, see the Control Function (CF) Package Users' Guide.

```
TF_INPUT
TF_ID  'Bills function' 1.
TF_TAB  4  !NPAR  X      Y
            1     0.0    1.0
            2     5.0    100.5
            3     100.8  55.
            4     5000.  90.993
!
TF_ID 'Sams constant function' 1. 0.
TF_TAB 1 !NPAR  X      Y
          1     0.     1.
...
```

```
!
! Note the next two functions illustrate two different
! ways of constructing a step function.
! These tabular functions with the control function can be
! used as a valve to control opening and closing.
!
CF_INPUT
!      CFNAME          CFTYPE
CF_ID 'valve control'   HYST
!      CFSCAL  CFADCN  CFVALR(INITIAL VALUE)
CF_SAI 1.      0.      0.0
!      MISCELLANEOUS
CF_MSC TF 'valve open' TF 'valve close'
CF_ARG 1 !NARG  CHARG         ARSCAL  ARADCN
!      look at pressure in control volume CV210
          1     CVH-P(CV210)  1.0     0.0
!
TF_INPUT
TF_ID  'valve open'  1.  0.
TF_BND  CONST-BND  CONST-BND ! this is the default
TF_TAB  2 ! NPAR  X      Y
             1  7.07E6 0. ! closed until p increases to 7.07 MPa
             2  7.07E6 1. ! full open
!
TF_ID  'valve close'  1.  0.
TF_BND  ERR-BND    ERR-BND
TF_TAB 4 ! NPAR X      Y
            1 -1.E20   0.
            2  6.7E6   0. ! opened until p decreases to 6.7 MPa
            3  6.7E6   1. ! full close
            4  1.E20   1.
```

Transfer Process (TP) Package
Users' Guide

The MELCOR Transfer Process (TP) package provides a standard interface for other MELCOR packages to send mass and its associated enthalpy to one or more packages. The material identities may change across the TP package.

This users' guide provides a short description of the TP package, including the default identity translation matrix and the user input requirements for MELGEN and MELCOR, respectively, with some examples.

Contents

LIST OF TABLES **2**

1 INTRODUCTION **3**

2 DEFAULT TRANSLATION MATRICES **5**

3 MELGEN USER INPUT **5**

TP_IN – *In* Transfer Processes and Connection to External Data Files. 5
TP_OUT – *Out* Transfer Process Record 8
TP_MTX – Translation Matrix 9

4 MELCOR USER INPUT **10**

TP_OUT – *Out* Transfer Processes Record 10
TP_MTX – Translation Matrix 11

5 EXAMPLE INPUT **12**

5.1 Example 1 12
5.2 Example 2 13

List of Tables

Table 3.1 Relationship between in and out transfer processes 6
Table 3.2 Relationships between in transfer process and associated EDF file. 7

1 Introduction

The MELCOR Transfer Process (TP) package provides a standard interface for a physics package to transfer mass and energy to one or more MELCOR packages. The TP package, in essence, functions as a sophisticated bookkeeper for inter-package mass and energy transfers. (The Control Volume Hydrodynamics [CVH] package provides its own utility interface for control volume fluid mass and energy transfers involving MELCOR packages; the TP package is not involved.)

There are two types of transfer processes, in and out. An *in* transfer process sends information to the TP package where it is stored. On request, the TP package retrieves the information, processes it, and sends it to a calling *out* transfer process. User input defines the relationships between the *in* and *out* transfer processes. Currently, required input to the TP package is prescribed by the Core (COR), Cavity (CAV), and RadioNuclide (RN) packages, and optionally by the Fuel Dispersal Interactions (FDI), Heat Structures (HS), and Containment Sprays (SPR) packages. (See the respective users' guide for details.) Eventually, much of this transfer of information will become transparent to the user and obviate the need for TP input.

Masses may change identities across the TP package. This capability was originally intended for uses such as conversion of steel and steel oxide (in the COR package) into iron, chromium, nickel, carbon, and their oxides (in the CAV package). This particular use has been eliminated by inclusion of the material composition information (specified via COR package input) in a separate TP interface array that is used directly by the CAV package to break the steel and steel oxide into their constituent components. This will allow definition of time-dependent steel compositions sometime in the future. The material conversion feature of the TP package is now used only to handle imperfect interfaces between packages or between a MELCOR package and an external data file.

The user must specify a translation matrix to relate materials in the *out* process to those in the *in* process through the matrix equation

$$\text{mass out} = \text{translation matrix} \times \text{mass in}$$

The total mass of a material is conserved only if the elements in all translation matrix columns for the material sum to one. Thus, it is possible for mass to be retained or created by the TP package, which might be necessary because of imperfect interfaces between packages. For example, if the COR package ejects B_4C and the CAV package does not recognize that material (as is currently the case), it must be retained by the TP package by using the translation matrix to eliminate the B_4C from the output masses.

Any mass retained or created by the TP package is printed in MELCOR output in terms of the *in* process masses.

										UO_2
UO_2		1	0	0	0	0	0			Zr
Zr		0	1	0	0	0	0			Steel
Steel	=	0	0	1	0	0	0		**X**	ZrO_2
ZrO_2		0	0	0	1	0	0			St. Ox.
St. Ox.		0	0	0	0	1	0			B_4C

The translation matrix may also be used to accommodate *in* processes defined by external data files that do not have the precise format expected by the *out* process. For example, a seven-group (Reactor Safety Study, WASH-1400) source of fission products could be partitioned into the default 15 groups expected by the RN package through an appropriate translation matrix with 15 rows and 7 columns.

Noble gases		1	0	0	0	0	0	0		Noble Gases
Alk. Metals		0	0	1	0	0	0	0		Halogens
Alk. Earths		0	0	0	1	0	0	0		Alk. Metals
Halogens		0	1	0	0	0	0	0	**X**	Alk. Earths
Chalcogens		0	0	0	0	.967	0	0		Te Group
Platinoids		0	0	0	0	0	.53	0		Trans. Metals
Trans. Metals		0	0	0	0	0	.47	.002		Lanthanides
Tetravalents	=	0	0	0	0	0	0	.702		
Trivalents		0	0	0	0	0	0	.296		
Uranium		0	0	1	0	0	0	0		
As, Sb		0	0	0	0	.033	0	0		
Sn, Ag		0	0	0	0	0	0	0		
Boron		0	0	0	0	0	0	0		
Water		0	0	0	0	0	0	0		
Concrete		0	0	0	0	0	0	0		

The total enthalpy of materials transferred from one package to another via the TP package may not be conserved, either because masses may not be conserved or because of differences in equation-of-state representations between the two packages. The out package that receives masses also receives thermodynamic variables to calculate the enthalpy of the masses it has just received and transmits that information back to the TP package. For example, the COR package may use temperature for the thermodynamic variable and calculate an enthalpy of 2 MJ for a mass. The CAV package may receive the mass and temperature but calculate an enthalpy of 2.7 MJ. The CAV package transmits the 2.7 MJ value back to the TP package, which will log an enthalpy difference of 0.7 MJ. The enthalpy differences are also printed in MELCOR output.

If the equations of state for a material are the same for two packages except for a difference in reference point, there will be no real energy error if temperature is used as the thermodynamic variable. If the equations of state are not the same, there is an energy discrepancy that cannot be resolved (without choosing one equation or the other as correct), and there will be either an energy error if temperature is held constant during the

transfer or a temperature jump if energy is held constant during the transfer. For the former case, the energy error will be on the order of the uncertainty in the equation of state (hence the difference in state formulations) as long as the temperature at which material is transferred is not on one side of a phase transition for one equation and on the other side of the phase transition for the other equation. MELCOR currently uses this approach to avoid step changes in temperature during debris ejection from the core to the cavity.

2 Default Translation Matrices

The commonly used identity translation matrix is available for use as default matrix number 1. It is square, and all its elements are zero except for those along the diagonal. This matrix preserves all characteristics of the incoming material when it is passed to the *out* transfer process.

3 MELGEN User Input

The user defines the *in* and *out* transfer processes and the matrices that convert the in materials to the out materials. Every *in* process must be associated with at least one *out* process. Every translation matrix must be associated with at least one *out* process.

The following input descriptions are generic in nature. Several MELCOR packages (COR, FDI, CAV, and RN packages) have specific requirements for TP package input. Details regarding the specific input required for these packages may be found in the respective users' guide.

The following TP input records must be preceded by a TP_INPUT record.

TP_IN – *In* Transfer Processes and Connection to External Data Files.

Required

This record defines all in transfer processes.

(1) NTPIN
Number of *in* transfer processes.
(type = integer, default = none, units = none)

The following data are input as a table with length NTPIN:

(1) NTP
In transfer process index.
(type = integer, default = none, units = none)

(2) NAMETPIN

In transfer process name.

(type = character*16, default = none, units = none)

(3) TYPE

Source of data used by this *in* transfer process (package(s) or external data file [EDF]):

(a) HS
(b) FDI
(c) RNFDI
(d) COR
(e) RNCOR
(f) EDF

The masses and the thermodynamic variables will be in the order defined by the input package (see Tables 3.1, 3.2 and discussion).

(type = character*5, default = none, units = none)

Table 3.1 Relationship between *in* and *out* transfer processes

SOURCE		RECEIVER	NMSIN	NMSOT	NTHRM	MTX
COR	→ TPIN $\xrightarrow{MTX}$ TPOUT →	FDI	6	5	9	6 x 5
RNCOR		RNFDI	NUMCLS	NUMCLS	1	DEF1
COR		CAV	6	5	9	6 x 5
RNCOR		RNCAV	NUMCLS	NUMCLS	1	DEF1
FDI		CAV	5	5	9	DEF1
RNFDI		RNCAV	NUMCLS	NUMCLS	1	DEF1
HS		SPR	1	1	1	DEF1

If RN package is active, the second transfer process for associated radionuclides (in addition to the one for debris) must be defined. It will be NMSIN = NUMCLS (described for input record RN1_DIM) and NTHRM = 1. Each record in the file will contain NUMCLS + 2 channels (dependent variables) because a cumulative enthalpy will also be written.

<u>Connection to External Data File</u>

The following optional fields allow association of an *in* transfer process with an external data file. (See the EDF Package Users' Guide.) It may be used to write out a history of the *in* process; this allows data from one MELCOR run to be saved for later use. Conversely, it may be used to read in such a history and to generate appropriate mass transfers from it; this allows data from a previous MELCOR run (or those from another simulation code) to be used to generate sources of debris or radionuclides as though they had been received through an *in* transfer process.

Table 3.2 Relationships between *in* transfer process and associated EDF file.

<table>
<tr><td colspan="6">EDF file is used to write out a history file: DIRECTION = PUSH</td></tr>
<tr><td>SOURCE</td><td></td><td>RECEIVER</td><td>NCHAN =</td><td>MTX</td><td>NOTE</td></tr>
<tr><td>COR</td><td rowspan="5">→ TPIN →</td><td rowspan="5">EDF</td><td rowspan="2">NMSIN + NTHRM + 1</td><td rowspan="5">not used</td><td rowspan="5">NMSIN, NTHRM values are uniquely defined through associated source package (see Table 3.1) and have not to be input by the user)</td></tr>
<tr><td>FDI</td></tr>
<tr><td>HS</td></tr>
<tr><td>RNCOR</td><td rowspan="2">NUMCLS + 2</td></tr>
<tr><td>RNFDI</td></tr>
<tr><td colspan="6">EDF file is used to read in a history file: DIRECTION = READ</td></tr>
<tr><td colspan="2">SOURCE</td><td colspan="2"></td><td>RECEIVER</td><td>NCHAN =</td><td>MTX</td></tr>
<tr><td>COR</td><td rowspan="5">→ EDF</td><td colspan="2" rowspan="5">→ TPIN $\xrightarrow{MTX}$ TPOUT→</td><td>FDI
CAV</td><td rowspan="5">NMSIN + NTHRM + 1
NMSIN, NTHRM values are uniquely defined through associated source package (see Table 3.1) and have not to be input by the user)</td><td rowspan="5">Dimension is fixed. Matrix is default or user defined on the corresponding TP_MTX record. NMSOT is uniquely defined and has not to be input.</td></tr>
<tr><td>FDI</td><td>CAV</td></tr>
<tr><td>HS</td><td>SPR</td></tr>
<tr><td>RNCOR</td><td>RNFDI
RNCAV</td></tr>
<tr><td>RNFDI</td><td>RNCAV</td></tr>
<tr><td colspan="2" rowspan="2">EDF</td><td colspan="2" rowspan="2">→ TPIN $\xrightarrow{MTX}$ TPOUT→</td><td rowspan="2">PACKAGE</td><td rowspan="2">EDFNMSIN + EDFNTHRM + 1
Fields EDFNMSIN, EDFNTHRM are required, values are defined by the user.</td><td rowspan="2">Translation matrix is user defined and must be input on the corresponding TP_MTX record. NMSOT must be input</td></tr>
<tr></tr>
</table>

If a file is written (DIRECT = PUSH), each record will contain NMSIN cumulative masses, the cumulative enthalpy, and NTHRM thermodynamic variables (i.e., NMSIN + NTHRM + 1 EDF channels). See the description of NTPCOR and RNTPCOR on input record COR_TP or of TPINAM and RNTPINAM on input record FDI_LOC for debris transfers.

If a file is read (DIRECT = READ), it is assumed to have the same format, although differences in the number and order of masses from the MELCOR transfer process it replaces may be accommodated through proper definition of the translation matrix using TP_MTX input.

Whatever the case, if the EDF file is used, the number of channels (NCHAN on record EDF_CHN for that file) must be in the order defined by the package that will use this *in* process.

(4) DIRECT

Direction of connection. This may be

(g) READ

To read a file

(h) PUSH
to write a history file

This field is required if EDF file is used.

(type = character*4, default = none, units = none)

(5) IP2EDF

Name of EDF file associated with this *in* transfer process.
This field is required if EDF file is used.

(type = character*16, default = none, units = none)

If TYPE = EDF and DIRECT = READ, the user must define the number of masses and the number of thermodynamic variables in the EDF file. In this case input of these fields is required, the corresponding translation matrix must be input on the TP_MTX record, and the number of masses NMSOT associated with this *out* transfer process must be defined on record TP_OUT.

(6) EDFNMSIN

User-defined number of masses in EDF file.
Required only if TYPE = EDF and DIRECT = READ.

(type = integer, default = none, units = none)

(7) EDFNTHRM

User defined number of thermodynamic variables in EDF file.
Required only if TYPE = EDF and DIRECT = READ.

(type = integer, default = none, units = none)

Example

```
TP_IN   6 !  NAMETPIN  NAMEPAC  DIRECT  IP2EDF  NMSIN NHTRM
           1  TP101      COR
           2  TP103      FDI
           3  TP601      RNCOR
           4  TP603      RNFDI
           5  TPinEDF    EDF      READ      EDF1      5     3
           6  TP201      COR      PUSH      EDF2
```

TP_OUT – *Out* Transfer Process Record

Required

This record defines *out* transfer processes, the associated *in* transfer process names and translation matrices, and, if needed, the number of masses associated with each *out* transfer process.

(1) NTPOUT

Number of *out* transfer processes.

(type = integer, default = none, units = none)

The following data are input as a table with length NTPOUT:

(1) NTP

Table sequencing index.

(type = integer, default = none, units = none)

(2) NAMETPOT

Name of *out* transfer process.

(type = character*16, default = none, units = none)

(3) NPOTOI

Name of associated *in* transfer process.

(type = character*16, default = none, units = none)

(4) OUTMTX

Name of associated translation matrix. Optional.
In process masses are multiplied by a translation matrix to generate the *out* process masses. Each *out* transfer process has a translation matrix associated with it. OUTMTX must be specified by the user-input name (refer to the TP_MTX input record described below) or by the character string DEF1 which specifies a default translation matrix (the identity matrix).

(type = character*16, default = DEF1, units = none)

(5) NMSOT

Number of masses associated with this *out* transfer process.
Number of masses associated with each *out* transfer process is uniquely defined through associated *in* transfer process if source of data (TYPE on appropriate TP_IN record) is package. Otherwise, the value must be input (TYPE = EDF).

(type = integer, default = none, units = none)

Example

```
TP_OUT  5 ! NAMETPOT ******* NPOTOI ********* OUTMTX *****NMSOT
        1   TP102            TP101            UIN103
        2   TP104            TP103            DEF1
        3   TP602            TP601            DEF1
        4   TP604            TP603            DEF1
        5   TPoutEDF         TPinEDF          MyMtx       4
```

TP_MTX – Translation Matrix

Required

This record defines the elements of a translation matrix. The input is in the form of a table.

(1) NAMEMTX

Name of translation matrix.

(type = character*16, default = none, units = none)

(2) NROW

Number of rows.

(type = integer, default = none, units = none)

(3) NCOL

Number of columns.

(type = integer, default = none, units = none)

Next NROW lines define values of matrix elements. First integer is the row index. The following NCOL real numbers are the NROW-th elements of the matrix.

Example

```
!****** NAMEMTX ******** NROW NCOL
TP_MTX UIN103            5  6
       !NROW   Matrix data as a table (NROW,NCOL)
        1      1.0  0.0  0.0  0.0  0.0  0.0
        2      0.0  1.0  0.0  0.0  0.0  0.0
        3      0.0  0.0  1.0  0.0  0.0  0.0
        4      0.0  0.0  0.0  1.0  0.0  0.0
        5      0.0  0.0  0.0  0.0  1.0  0.0
TP_MTX MyMTX             4  5
       !NROW   Matrix data as a table (NROW,NCOL)
        1      1.0  0.0  0.0  0.0  0.0
        2      0.0  1.0  0.0  0.0  0.0
        3      0.0  0.0  1.0  0.0  0.0
        4      0.0  0.0  0.0  1.0  0.0
```

4 MELCOR User Input

For a given *out* transfer process, the user may redefine the associated *in* transfer process name and translation matrix. The elements of a translation matrix may be redefined. No new *in* transfer processes, *out* transfer processes, or translation matrices may be defined in MELCOR. Every *in* process must be associated with at least one *out* process.

TP_OUT – *Out* Transfer Processes Record

Optional

For each of the existing *out* transfer processes, this record may redefine the associated *in* transfer process name and the associated translation matrix.

(1) NUMSTR

Number of table records. Each table record redefines data for one *out* transfer process.

(type = integer, default = none, units = none)

The following data are input as a table with length NUMSTR:

(1) NSTR

Table record index.

(type = integer, default = none, units = none)

(2) NAMETPOT

Name of *out* transfer process. This may not be changed in MELCOR.

(type = character*16, default = none, units = none)

(2) NPOTOI

Name of associated *in* transfer process. May be changed in MELCOR.

(type = character*16, default = none, units = none)

(3) OUTMTX

Name of associated translation matrix. Optional.
In process masses are multiplied by a translation matrix to generate the *out* process masses. Each *out* transfer process has a translation matrix associated with it. OUTMTX must be specified by the user-input name (refer to the TP_MTX input record described below) or by the character string DEF1, which specifies a default translation matrix (the identity matrix). Any redefinition of translation matrices must be done so that the number of rows and columns do not change, whether default or user-defined matrices are used.

(type = character*16, default = none, units = none)

(4) NMSOT

Number of masses associated with this *out* transfer process. This may not be changed in MELCOR.

(type = integer, default = none, units = none)

TP_MTX – Translation Matrix

Optional

This record allows the user to redefine the elements of the translation matrix and may be input in MELCOR for a matrix already defined in MELGEN. However, the number of rows and columns of a matrix cannot be changed in MELCOR.

(1) NAMEMTX

Name of translation matrix.

(type = character*16, default = none, units = none)

(2) NROW

Number of rows.

(type = integer, default = none, units = none)

(3) NCOL

Number of columns.

(type = integer, default = none, units = none)

Next NROW lines define values of matrix elements. First integer is the row index. The following NCOL real numbers are the NROW-th elements of the matrix.

5 Example Input

5.1 Example 1

Example 1 provides sample input for the COR, FDI, CAV, and TP packages for a typical plant calculation.

```
! COR, FDI, AND CAV PACKAGE INPUT
! NO INPUT REQUIRED FOR RN PACKAGE
COR_INPUT
! COR'IN'TP
COR_TP  COR_IN  RNCOR_IN   NO   NO
...
! NCV NCAV FDI'IN'TP FDI'OUT'TP
FDI_INPUT
FDI_ID   FDI5
FDI_LOC CV500 CAV CAV10 FDI_IN FDI_OUT RNFDI_IN RNFDI_OUT
...
CAV_INPUT
! CAV'OUT'TP
CAV_ID   CAV10
CAV_TP   CAV_OUT  RNCAV_OUT
...
!
TP_INPUT

TP_IN   1 !NAMETPIN *** NAMEPAC DIRECT IP2EDF *** NMSIN NHTRM **
        1  COR_IN           COR
        2  FDI_IN           FDI
        3  RNCOR_IN         RNCOR
        4  RNFDI_IN         RNFDI
TP_OUT  1 !NAMETPOT ******* NPOTOI ********* OUTMTX ***NMSOT****
        1  FDI_OUT          COR_IN          UIN506       5
        2  CAV_OUT          FDI_IN          DEF1         5
        3  RNFDI_OUT        RNCOR_IN        DEF1         16
        4  RNCAV_OUT        RNFDI_IN        DEF1         16
! COR-FDI TRANSLATION MATRIX
! *** NOTE *** CONTROL POISON MASS IS NOT CONSERVED
! NROW NCOL
```

```
 !***** NAMEMTX ******** NROW NCOL
TP_MTX UIN506            5  6
       !NROW   Matrix data as a table (NROW,NCOL)
        1      1.0   0.0   0.0   0.0   0.0   0.0 !  UO2 MASS
        2      0.0   1.0   0.0   0.0   0.0   0.0 !  ZR MASS
        3      0.0   0.0   1.0   0.0   0.0   0.0 !  STEEL MASS
        4      0.0   0.0   0.0   1.0   0.0   0.0 !  ZRO2 MASS
        5      0.0   0.0   0.0   0.0   1.0   0.0 !  STEEL OXIDE
                                                   !  MASS
```

5.2 Example 2

Example 2 provides sample input to split ejected core debris into two separate reactor cavities for molten core-concrete interactions. Seventy-five percent of the debris is discharged to cavity name CAV2, and the remainder is discharged to cavity name CAV3.

```
! COR AND CAV PACKAGE INPUT
COR_INPUT
! COR'IN'TP
COR_TP  COR_IN   RNCOR_IN   NO   NO
...
CAV_INPUT
! CAV2'OUT'TP
CAV_ID CAV2
CAV_TP   CAV2_OUT   RNCAV2_OUT
...
! CAV3'OUT'TP
CAV_ID CAV3
CAV_TP   CAV3_OUT  RNCAV3_OUT
...
TP_INPUT

TP_IN   1 !NAMETPIN *** NAMEPAC DIRECT IP2EDF *** NMSIN NHTRM **
        1  COR_IN             COR
        2  RNCOR_IN           RNCOR
TP_OUT  1 !NAMETPOT ******* NPOTOI ********* OUTMTX ***NMSOT****
        1  CAV2_OUT         COR_IN          UIN2         5
        2  CAV3_OUT         COR_IN          UIN3         5
        3  RNCAV2_OUT       RNCOR_IN        UIN12        15
        4  RNCAV3_OUT       RNCOR_IN        UIN13        15
!
! COR-CAV TRANSLATION MATRICES
! 0.75 TO CAVITY 2, 0.25 TO CAVITY 3
! *** NOTE *** CONTROL POISON MASS IS NOT CONSERVED
! NROW NCOL
TP_MTX UIN2             5  6 ! 5X6 MATRIX
 ! NROW   Matrix data as a table (NROW,NCOL)
    1   0.75  0.0   0.0   0.0   0.0  0.0 ! non-zero UO2 MASS
    2   0.0   0.75  0.0   0.0   0.0  0.0 ! non-zero ZR MASS
    3   0.0   0.0   0.75  0.0   0.0  0.0 ! non-zero STEEL MASS
    4   0.0   0.0   0.0   0.75  0.0  0.0 ! non-zero ZRO2 MASS
    5   0.0   0.0   0.0   0.0   0.75 0.0 ! non-zero STEEL OXIDE
                                         ! MASS

TP_MTX UIN3             5  6 ! 5X6 MATRIX
 ! NROW   Matrix data as a table (NROW,NCOL)
    1   0.25  0.0   0.0   0.0   0.0  0.0 ! non-zero UO2 MASS
    2   0.0   0.25  0.0   0.0   0.0  0.0 ! non-zero ZR MASS
    3   0.0   0.0   0.25  0.0   0.0  0.0 ! non-zero STEEL MASS
    4   0.0   0.0   0.0   0.25  0.0  0.0 ! non-zero ZRO2 MASS
    5   0.0   0.0   0.0   0.0   0.25 0.0 ! non-zero STEEL OXIDE
                                         ! MASS
```

```
TP_MTX UIN12             15  15 ! 15X15 MATRIX
! NROW   Matrix data as a table (NROW,NCOL)
 1  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 2  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 3  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 4  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 5  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 6  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.  0.
 7  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.  0.
 8  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.  0.
 9  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.  0.
10  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.  0.
11  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.  0.
12  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.  0.
13  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.  0.
14  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75 0.
15  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.75

TP_MTX UIN13             15  15 ! 15X15 MATRIX
! NROW   Matrix data as a table (NROW,NCOL)
 1  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 2  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 3  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 4  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 5  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 6  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.  0.
 7  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.  0.
 8  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.  0.
 9  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.  0.
10  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.  0.
11  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.  0.
12  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.  0.
13  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.  0.
14  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25 0.
15  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.25
```