NUREG/CR-3905
ORNL/NSIC-223

Sequence Coding and Search System for Licensee Event Reports

User's Guide

Prepared by N. M. Greene, G. T. Mays/ORNL
M. P. Johnson/JBFA

Oak Ridge National Laboratory

Prepared for
U.S. Nuclear Regulatory
Commission

GPO Printed copy price: $6.00

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D. C. 20555

AUG 28 1984

MEMORANDUM FOR: Harold R. Denton, Director, NRR
Robert B. Minogue, Director, RES
Richard C. DeYoung, Director, IE
John G. Davis, Director, NMSS
Regional Administrators

FROM: C. J. Heltemes, Jr., Director
Office for Analysis and Evaluation
of Operational Data

SUBJECT: AVAILABILITY OF COMPUTERIZED DATA BASE ON LICENSEE EVENT REPORT (LER) INFORMATION - SEQUENCE CODING AND SEARCH SYSTEM (SCSS) (NUREG/CR-3905)

AEOD has contracted with the Nuclear Operations Analysis Center (NOAC) at the Oak Ridge National Laboratory (ORNL) for the development and operation of the Sequence Coding and Search System (SCSS), a computerized data base on Licensee Event Report information. Work has progressed such that the file is now available for direct on-line access by other offices within the NRC.

Based upon a review of the enclosed material (review of Chapters 1, 2, and 3 of the enclosed guide should provide a good understanding of the system's capabilities), please let us know:

1. Does your office desire direct access to the Sequence Coding and Search System (SCSS)? As indicated in the enclosure, the file contains data on operating experiences at nuclear power plants in operation in the U.S. If your office desires direct access, please provide AEOD with the name(s), organization, mail stop, and phone number of the person(s) assigned to access the system.

   AEOD will:

   a. Provide a controlled copy of the user's guide and additional system documentation.

   b. Work with RM to obtain user identification codes and passwords.

   c. Provide training, if needed, to system users.

2. Are the existing services provided by NOAC and AEOD sufficient for your office needs? NOAC can be contacted directly and searches of the data can be made expeditiously with responses provided overnight via express mail service, if necessary. AEOD can also perform searches and has local printout facilities such that same-day or overnight responses can usually be provided. Due to resource limitations, however, we request that you normally use the service provided by NOAC and request our assistance when an immediate response is necessary (i.e., less than 1 day) or in the event of difficulty.

If you have any questions or wish to discuss details on the SCSS data base, please contact Eugenia Boyle of my staff at 492-4498.

C. J. Heltemes, Jr., Director
Office for Analysis and Evaluation
of Operational Data

Enclosure:
Sequence Coding and Search
System User's Guide

cc w/enclosure:
W. Dircks, EDO
J. Roe, DEDO
T. Rehm, AO/EDO
V. Stello, DEDROGR
J. Sniezek, DEDROGR
R. Scroggins, RM
J. Buchanan, NOAC
N. Greene, ORNL
A. Malinauskas, ORNL
H. Trammell, ORNL
W. Parker, DOE
W. Bibb, DOE
Recipients of the Licensee Event Report
(LER) Compilation

NUREG/CR-3905
ORNL/NSIC-223

Sequence Coding and Search System for Licensee Event Reports

User's Guide

Manuscript Completed: August 1984
Date Published: August 1984

Prepared by
N. M. Green, G. T. Mays, Oak Ridge National Laboratory
M. P. Johnson, JBF Associates, Inc.

Nuclear Operations Analysis Center
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Subcontractors:
Science Applications, Inc.
800 Oak Ridge Turnpike
Oak Ridge, TN 37830

JBF Associates, Inc.
Technology Drive
1000 Technology Park Center
Knoxville, TN 37922

Prepared for
Office for Analysis and Evaluation of Operational Data
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN A9451

ABSTRACT

The Nuclear Regulatory Commission's Office for Analysis and Evaluation of Operational Data has developed, through the Nuclear Operations Analysis Center (NOAC) at Oak Ridge National Laboratory (ORNL), a system to aid in the evaluation of the Licensee Event Reports (LERs) submitted by the nuclear power plant utilities. The primary objective of the Sequence Coding and Search System (SCSS) is to reduce the descriptive text of the incident reports to a coded sequence that is both computer-readable and computer-searchable. This system provides a structured format for detailed coding of component, system, and unit effects, as well as personnel errors. The database contains all current LERs submitted by the nuclear power plant utilities after January 1, 1981, and is updated on a continual basis with new LERs, as they are submitted. The database is maintained by NOAC on the IBM-3033 computer system at ORNL. Following a description of SCSS and structure of the database, a tutorial section is provided to acquaint the first-time user with logon procedures and the necessary commands to retrieve, display, and analyze LERs. Each command is subsequently discussed in detail in the fundamental and advanced command sections.

CONTENTS

1. DESCRIPTION OF THE DATABASE

1. DESCRIPTION OF THE DATABASE

1. DESCRIPTION OF THE DATABASE

The Sequence Coding and Search System (SCSS) database contains all current Licensee Event Reports (LERs) submitted by the nuclear power plant utilities after January 1, 1981. It is updated on a continual basis with new LERs, as they are submitted. The database was developed by the Nuclear Regulatory Commission's (NRC's) Office for Analysis and Evaluation of Operational Data (AEOD) through the Nuclear Operations Analysis Center (NOAC) at Oak Ridge National Laboratory (ORNL) and is maintained by the Nuclear Safety Information Center (NSIC)--a part of NOAC--on the IBM-3033 computer system at ORNL. The SCSS database employs the JOSHUA system along with new software explicitly written to manage the database.

The primary objective of SCSS is to reduce the descriptive text of the incident reports to a coded sequence(s) that is both computer-readable and computer-searchable. This system provides a structured format for detailed coding of component, system, and unit effects, as well as personnel errors. The SCSS database uses coded information extensively, and the codes are non-mnemonic. For instance, a system code of AA designates the reactor core, and AB designates the control rod drive system in a PWR. More than 6000 four-character codes designate component vendors. The reactors are identified by three-digit docket codes.

NOTE: Listings of the codes used in SCSS are available in Sequence Coding and Search System Coder's Manual for Licensee Event Reports - Code Listings, ORNL/NSIC-189. Many users, however, will receive all the help they need with the codes by using the AUTH, DOCKET, and FIND HELP commands explained in Chap. 3 of this manual.

SCSS CAPABILITIES

Operating experience data from nuclear power plants are essential for safety and reliability analyses. The LERs submitted to the NRC by the nuclear power plant utilities contain much of these data.

Historically, LERs were searched and retrieved primarily by using descriptive keywords. This method, however, limited the adaptability of the database for detailed information searches. Also, the size of the database has increased significantly in the past decade. To overcome these two drawbacks, the SCSS was developed.

Features of SCSS include:

- o sophisticated and efficient search techniques,
- o information from the LER supplemental data sheets for pre-1984 LERs,
- o information from the LER abstract and narrative descriptions for post-1984 LERs,
- o component identification,
- o codes that categorize components in both fine (specific--check valve) and coarse (generic--valves) detail,
- o detailed descriptions of personnel actions,
- o identification of the loss of one or more trains and/or systems,
- o identification of multiple initiators of an event (if they exist), and
- o information regarding unit effects and radiological releases and exposures.

ABOUT THIS MANUAL

This manual is both a user's guide and a reference manual. If you are using SCSS for the first time, you should familiarize yourself with its capabilities and structure, as detailed in Chap. 1. This chapter contains essential information about the sequence coding concept and about the structure of the database. Then you should review Chap. 2, "Tutorial--Retrieving, Displaying, and Analyzing LER Information." This chapter will give you hands-on experience in logging on to the system and executing basic commands. The chapter also includes sample output from the system.

Once you have a general knowledge of SCSS (as described later in this chapter), you should study Chap. 3 in detail. Chapter 3 describes the function, syntax, and variations or options of the fundamental commands used in the SCSS. It also discusses the HELP capabilities of the system, which provide both general on-line assistance and assistance in using the database codes.

After you have become familiar with the SCSS, you may want to review Chap. 4, "Advanced Features." However, we suggest that the novice SCSS user not proceed beyond Chap. 3. The commands in Chap. 4 are more complex and are not needed to perform most searches of interest to a novice user.

SEQUENCE CODING CONCEPT

The SCSS database is designed to meet two system requirements: (1) that all relevant technical information from the LER form (NRC form 366) and from supplemental information are encoded and (2) that all technical information is sufficiently tagged for precise retrieval. So that these requirements could be met, the sequence coding concept was applied.

What Information is Collected

The information collected includes all data needed to completely describe the event and the information to identify the LER. Sequence coding requires that the technical reviewer first identify individual occurrences that make up the sequences as reported within an LER (see Fig. 1) then encode each piece of information using a computer-readable and -searchable code. Each individual occurrence usually constitutes a step. (Steps are discussed in "Structure of the Database" later in this section.) Each event, then, is ordered in time. The coding indicates each cause, system and component involved, manufacturer or vendor, and failure or fault effect. The SCSS database also records personnel actions, but handles them in a manner similar to equipment failure. For personnel actions, the system attribute identifies the type of activity the person is involved in--maintenance, design, installation, etc.; the component attribute identifies the type of personnel involved--licensed operator, contractor, etc. Please note: no information on corrective action is collected.

Coded Step Matrix

The information about each LER, when coded, is displayed in a coded step matrix; see the sample LER record shown in Fig. 2. The coded step matrix is a systematic way of recording the occurrences that took place during an event. Three special linkage fields (STEP, LK, and SLK) describe how the the individual occurrences (horizontal rows of the matrix) describe the characteristics (e.g., system involved, cause) of each occurrence. When decoded, the matrix explains:

- the steps (in order) involved in an event and any applicable links between steps,
- the cause of each component failure, system failure, or personnel action,
- the primary system involved in the occurrence (with the interfacing system if applicable),
- the type of component involved,
- the component descriptors such as vendor and number of components that failed,
- information on instantaneous, preexisting, or potential failures,

FIGURE 1. RELATIONSHIP OF EVENT, SEQUENCE, AND OCCURRENCE

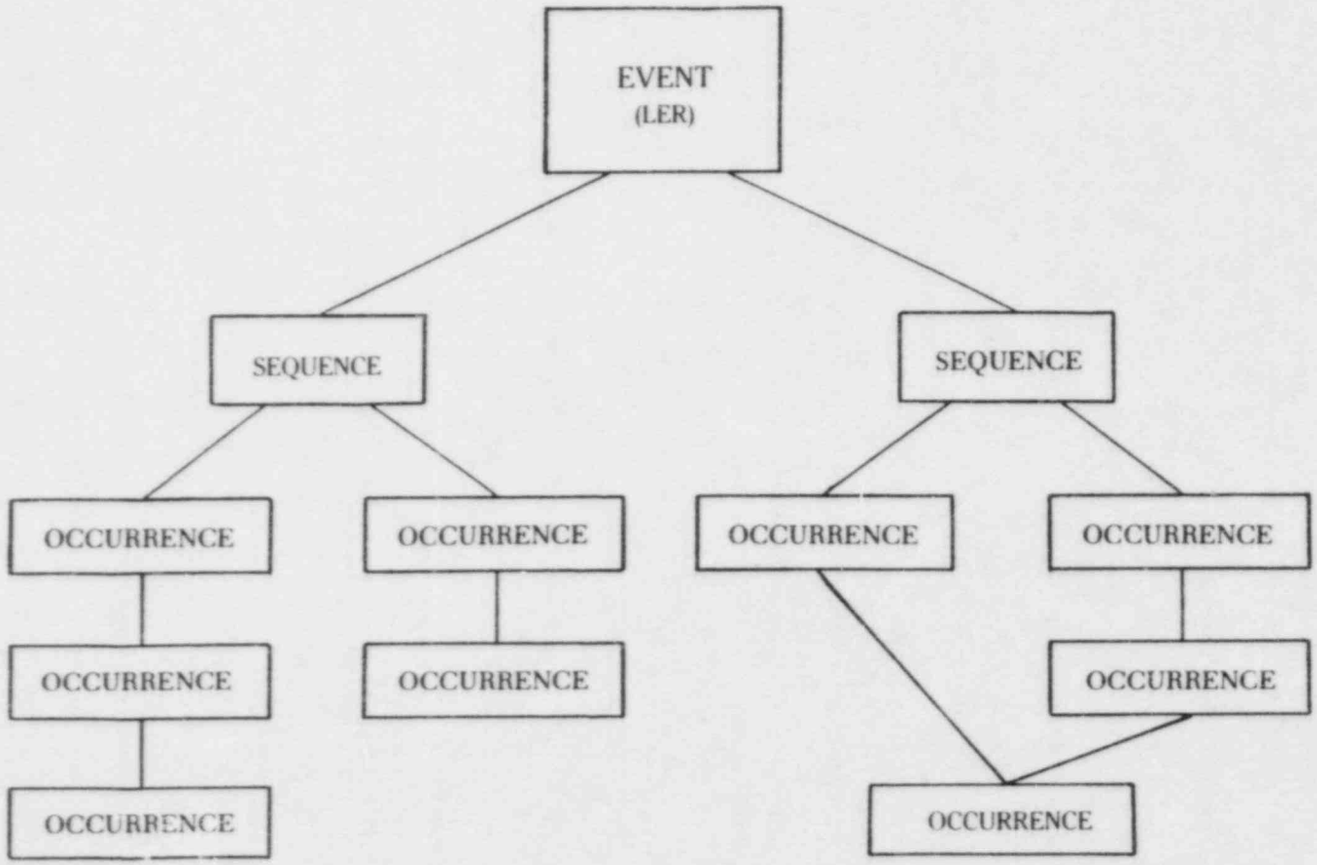


FIGURE 2. AN EXAMPLE OF A SINGLE LER DATABASE RECORD

Header Information

DOCKET	YEAR	LER NUMBER	REVISION	DCS NUMBER	NSIC	EVENT DATE
293	1984	005	0	8405080272	189610	4-4-1984

Comments

COMMENTS
VALVES MODEL #7567F. STEP 2: COMP XVZ = PILOT VALVE.

Docket Information

DOCKET: 293 PILGRIM 1 TYPE: BWR
REGION: 1 NSSS: GE
ARCHITECTURAL ENGINEER: BECH
FACILITY OPERATOR: BOSTON EDISON CO.
SYMBOL: AXA

Watch-List Codes

WATCH-LIST CODES FOR THIS LER ARE:
913 UPDATE NEEDED

Reportability Codes

REPORTABILITY CODES FOR THIS LER ARE:
10 10 CFR 50.73(a)(2)(i): Shutdowns or technical specification violations.

Reference LERs

REFERENCE LERS:

1 293/81-062

Coded Step Matrix

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF
1	0		PD	BR		VLVS		2	1		1	M	T	L	DC
2	1		RC	BR		XVZ	T020	1	1		1	A	TR	L	KF
3	2		RC	BR		ORVZ	T020	2	1		1	A	TR	L	KB
4				XX								H	XX		YC
5				YY								N	N		YC

Abstract

ABSTRACT
POWER LEVEL - 000%. ON 4/4/84, DURING A REFUELING OUTAGE, THE MAINTENANCE DEPARTMENT WAS NOTIFIED BY WYLE LABORATORIES THAT THE PILOT VALVES ON TWO OF THE TARGET ROCK TWO-STAGE SAFETY RELIEF VALVES (S/RV'S) DID NOT LIFT WITHIN SPECIFICATION WHEN DIAGNOSTICALLY TESTED IN THE AS-FOUND CONDITION. THIS IS CONTRARY TO THE REQUIREMENTS OF THE INTENT OF PNPS TECH SPEC 2.2.B WHICH REQUIRED THE S/RV'S TO LIFT AT 1095 PSI PLUS OR MINUS 11 PSI. THE MOST PROBABLE CAUSE OF THE SAFETY RELIEF VALVES NOT LIFTING HAS INITIALLY BEEN DETERMINED TO BE STUCK PILOT VALVES. DETERMININATION OF ROOT CAUSE AND CORRECTIVE ACTION IS PENDING FURTHER ANALYSIS AND TESTING.

- o the type of performance (i.e., whether the system, component, or personnel failed totally, partially, or not at all),
- o the detection method of the component or system failure or the personnel action,
- o the effect of the individual occurrence,
- o the effect of the sequence on the unit, and
- o information describing any radioactive releases or personnel exposures.

Other Information Contained in an LER

As Fig. 2 shows, each LER record contains as many as seven other categories of information in addition to the coded step matrix. The header information, comments, docket information, watch-list codes, reportability codes (beginning with 1984 LERs), reference LERs, and abstract are explained in "Structure of the Database."

Search and Retrieval Capabilities

The sequence coding concept provides the following search and retrieval capabilities:

- o selection of component failure data,
- o selection of system failure data,
- o determination of the failure's effect on the power plant unit,
- o selection of trend and pattern information,
- o identification of predetermined failure characteristics, and
- o determination of relational aspects of individual occurrences.

SCSS allows for very simple to complex searches involving the interrelationships among coded fields and steps. The system can also provide statistical information that supports evaluation of trends and patterns and a means for identifying relational aspects of occurrences, after a selected group of LERs has been captured. Other analysis aids in SCSS include graphical representations of sequential relationships of individual occurrences in an event-tree format; two-dimensional tables that represent relationships among various values for any two fields; and trend charts for specified field values.

STRUCTURE OF THE DATABASE

Figure 2 is an example of a typical LER record. Each LER contains all the data elements needed to completely describe the event. These data elements are grouped in functional categories and described below. Please refer to the example LER (Fig. 2) when reviewing the description of the data elements. Note that most of the data are entered in code; however, output options permit automatic decoding of coded information if desired. The data elements explained in "Coded Step Matrix" are from step 1 of the example LER.

Header Information

Docket	293	Docket number
Year	1984	Year of event
LER Number	005	LER number
Revision	0	Revision number
DCS Number	8405080272	NRC Document Control System (DCS) number
NSIC	189610	Nuclear Safety Information Center (NSIC) accession number
Event Date	4-4-1984	The month, day, and year the event occurred

Comments Given by coder when extra information is available or when the event contains an unusual or complex characteristic that cannot be adequately described via coding

Docket Information

Docket	293 Pilgrim 1	Docket number and power plant name
Type	BWR	Nuclear facility type
Region	1	NRC region number
NSSS	GE	Nuclear Steam Supply System (NSSS) vendor--General Electric
Architectural Engineer	BECH	Plant architectural engineer code--Bechtel
Facility Operator	Boston Edison Co.	Utility operating the power plant
Symbol	AXA	Power plant utility code

Watch-List Codes 913 Watch-list codes supplied by coder--913 for "update needed" (see description on p. 1-9)

Reportability Codes 50.73 (a)(2)(i) Reportability requirement under which the LER was submitted--10 CFR 50.73(a)(2)(i)

Reference LERs 293/81-062 Docket, year, and LER number of previous, similar events

Coded Step Matrix

STEP	1		Step - The step number order in which occurrences took place.
LK	0		Link - The link number relates the object step to prior steps.
SLK			Sublink - The sublink entry (an alphabetic code) links multiple occurrences that produce other occurrences.
CAUSE	PD	Crud buildup	Cause - Cause of component or system failure or personnel action.
PSYS	BR	Nuclear boiler overpressure protection (BWR)	Primary System - Primary system involved, the system in which the component belongs, or the type of personnel activity involved.
ISYS			Interfacing System - Interfacing system involved, such as the system being controlled or monitored, the system being isolated by the containment isolation system, or the system that is leaking to or from the primary system.
COMP	VLVS	Valve seat	Component - The component that failed, the trains or channels of a system that were affected, indication of total system failure, the type of personnel action, or other units affected by the event.
VEND			Vendor - Name of the vendor of the component that failed.
QUAN	2		Quantity - Number of failed components, trains, channels, or personnel actions.
TR	1		Train - Identifies failures in the sequence that belong in the same or different train(s).
CH			Channel - Identifies failures in the sequence that belong in the same or different channel(s).
DI	1		Differ - Identifies whether the person or component in the object step is the same or different from the person or components of a previous step.
T	M	Actual pre-existing, undetected	Timing - Timing of the failure (instantaneous, preexisting, or potential), initial unit conditions when PSYS=XX, and radioactivity release to environment when PSYS=YY.

P	T	Total fault	Performance - Equipment performance (total failure, partial failure, or no failure, plus whether repair was required), effect on the unit when PSYS=XX, and personnel exposure when PSYS=YY.
D	L	Special test/ inspection	Detection - Method of detection.
EFF	DC	Bind/jam	Effect - Resulting effect of the failure.
ICOMP			Coarse Component Code - This is a code for a group of like components (for instance, a single ICOMP code for valves encompasses about 150 separate component codes that can be used for valves). This code is not included as part of standard output.
Abstract			A brief textual description of the event is provided by the licensee as part of the LER.

Watch-List Codes

Technical reviewers assign watch-list codes to certain events to identify specific characteristics of the events that (1) are not directly amenable to sequence coding or (2) may present difficulties in formulating a search strategy that will retrieve the information. (For example, LERs involving physical security problems, perhaps because unknown or unauthorized personnel removed equipment from service, would be difficult to categorize; such an event would be assigned a watch-list code of 810, which is "security considerations.") The watch-list codes enable these events to be retrieved directly. For a listing of available watch-list codes that are searchable, see Sequence Coding and Search System Coder's Manual for Licensee Event Reports - Code Listings, ORNL/NSIC-189.

Inverted Fields

Eleven of the fields in the database are inverted. The fields that are inverted can be searched for all LERs that contain a given code via tables constructed by the computer (as opposed to searching for the codes in the coded step matrix of each LER). The inverted fields allow you to locate a group of LERs very quickly. For example, when you ask the computer to FIND diesel engines in the COMP field, it will locate

and can report the docket, year, and LER number information for all LERs that contain the diesel engine code in the component field without really retrieving the full record for each LER.

The 11 inverted fields are:

CAUSE (cause)	P (performance)
PSYS (primary system)	D (detection)
ISYS (interfacing system)	EFF (effect)
COMP (component)	ICOMP (coarse component code)
VEND (component vendor)	WATCH (watch-list code)
T (timing)	

Types of Steps

The general way in which information is structured in the database is depicted in Fig. 3. The six types of steps are of particular importance in SCSS because each one contains certain identifying characteristics that relate to informational content. (Remember that a horizontal line on the coded step matrix constitutes a step.) The six steps are (1) equipment failure, (2) personnel action, (3) subsystem effect, (4) system effect, (5) unit effect, and (6) environmental effect. Figure 4, presented at the end of the discussion of the six step types, summarizes the usage of certain fields in the individual steps.

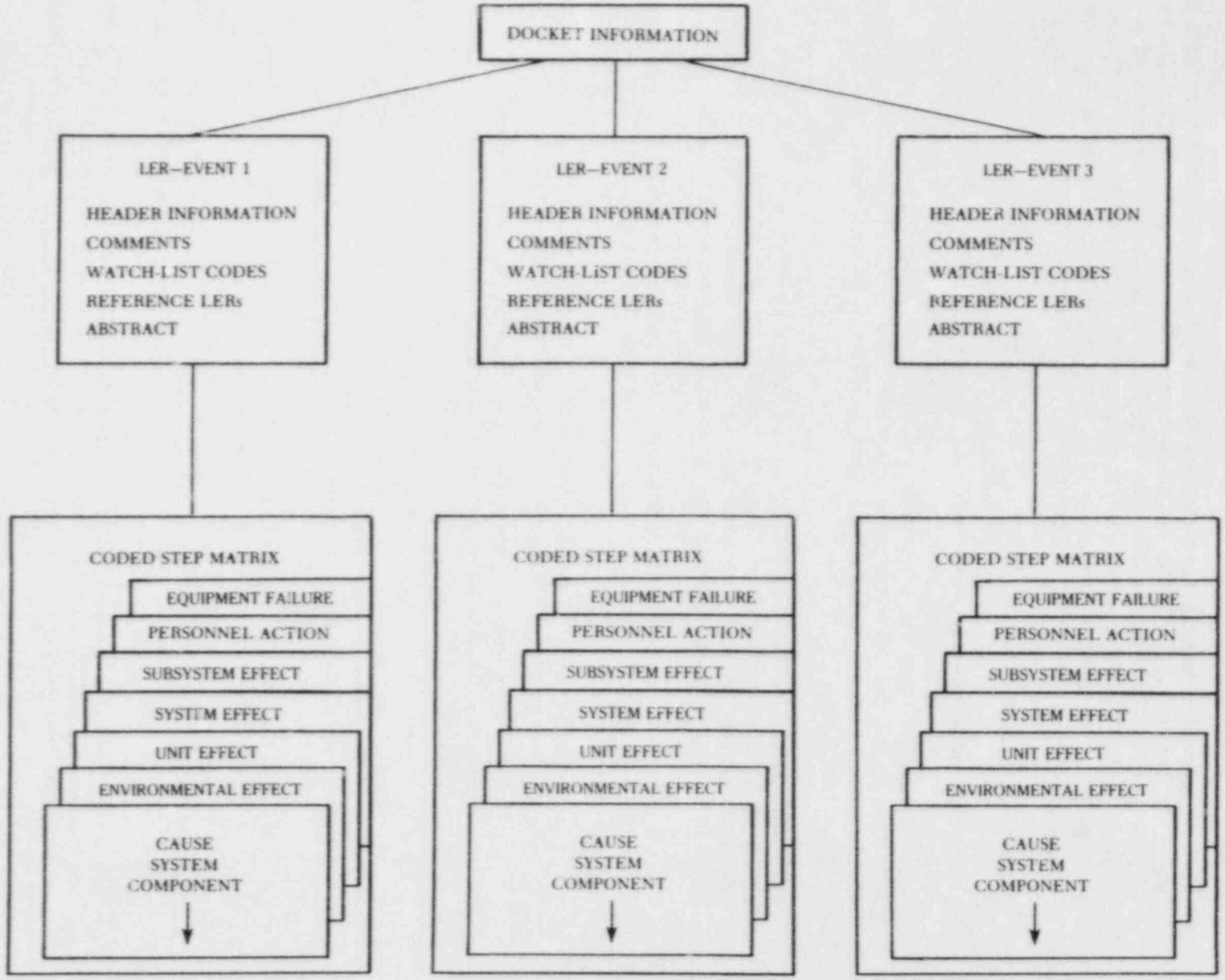
Equipment Failure

The most obvious type of occurrence is a failure of equipment (component). The equipment may have failed to perform its intended function because of (1) a reason(s) for which the equipment is held directly accountable (the pump failed to run because of wear) or (2) a command fault, in which the equipment failure was caused by anything other than failure of the component itself (a pump failed to start because it did not receive the signal to start).

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF	ICOMP
1			EF	CL	CB	HX	Y021	1	1		1	A	TR	I	HN	200

Decoded, this equipment failure step contains the following information. The cause of the equipment failure was corrosion/oxidation (CAUSE=EF). The principal system involved was the Emergency Generator Cooling System (PSYS=CL), and the component interfaced with the Essential Raw Cooling Water/Service Water System (ISYS=CB). The component involved was a heat exchanger (COMP=HX), sold by the Young Radiator Company (VEND=Y021). One heat exchanger (QUAN=1) was involved; it was in the first train (TR=1) of the Emergency Generator Cooling System mentioned in this LER. Values are supplied for the channel column only when the PSYS column contains codes beginning with "I" (indicating that an instrumentation system is involved). If the value of 1 for differ (DI=1) appears again in another step involving a heat exchanger, then the two steps involve the same heat exchanger. A different value (e.g., 2 or 3) in the differ column in another step involving a heat exchanger would indicate

FIGURE 3. SCSS DATABASE STRUCTURE



that different heat exchangers were involved. The failure, an actual immediate failure (T=A), was total and required corrective maintenance or repair (P=TR). The failure was detected during a routine test or inspection (D=I). The effect on the component was blockage of flow through the heat exchanger (EFF=HN). Finally, the component belongs to the generic class of "heat exchanger" (ICOMP=200).

Personnel Action

The coded sequences include personnel action steps when a personnel action initiates the sequence (e.g., the operator bumps a switch and trips a breaker) or the personnel action is an integral part of the sequence (e.g., in response to a component failure, the operator makes an incorrect action). Repair actions performed by personnel are not included in SCSS.

The distinguishing feature of personnel steps is that the PSYS code in the step always begins with "P". The personnel action step is coded in the same manner as the equipment failure step. However, the following attributes have special codes for personnel action steps:

- CAUSE (cause) - indicates the cause of the personnel action (e.g., inadequate training, disregard of requirement);
- PSYS (primary system) - indicates the type of activity the person was performing (e.g., maintenance, testing);
- COMP (component) - indicates the classification of the personnel involved (e.g., utility licensed operator, contractor);
- DI (differ) - indicates whether the personnel involved in that step are the same personnel described in a previous or subsequent step (the differ codes for personnel are alphabetic, as compared to numeric values for equipment steps);
- EFF (effect) - indicates the type of action taken (e.g., consequential omission of task, analysis, or step).

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF	ICOMP
2			SH	PM		PCP		1			A	M	T	I	TB	270

The above personnel action step decodes as follows. The cause of the personnel action was an error in requirements (CAUSE=SH). The person, who was performing a maintenance/repair activity (PSYS=PM), was a contractor employee (COMP=PCP). One person was involved (QUAN=1). If the value of A (DI=A) appears again in another step involving contractor personnel, then the two steps involve the same personnel. A different value (e.g., B or C) in the differ column would indicate the personnel in the two steps involving contractor personnel are not the same. The action occurred before the event, but had not been detected (T=M means that the action was preexisting). The error by the contractor personnel

was a total failure (P=T, the only allowable value for personnel steps) and was detected during a routine test or inspection (D=I). The effect of the error was a consequential omission of a task, analysis, or step (EFF=TB). Finally, the personnel code used in the component field belongs to the generic class of "personnel" (ICOMP=270).

Subsystem and System Failures

In addition to occurrences that involve individual components and personnel actions, occurrences can be associated with the entire system or an entire train of a given system (i.e., a subsystem). Like equipment occurrences, subsystem or system occurrences include a cause and effect, but the effect applies to an entire train of a system or to an entire system. Such occurrences are usually the result of one or more command faults, failures, or personnel actions associated with individual components. The subsystem and system occurrences use the same fields as the equipment and personnel occurrences, but use different codes.

Subsystem

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF	ICOMP
3			RS	EH		1X3		1	1		1	A	T	I	YB	930

Subsystem steps are characterized by the special use of the component field to indicate the number of trains in the system that failed. The subsystem step was caused by a previous step(s) (CAUSE=RS). The subsystem involved was part of the Emergency Power Generation System (PSYS=EH). This step involved one of the three trains of this system (COMP=1X3). The one train (QUAN=1) was the first train (TR=1) of the Emergency Power Generation System mentioned in this LER. If the value of 1 for differ (DI=1) appears again in another step involving this train of the Emergency Power Generation System, then the two steps involve the same train. It was an actual, immediate fault (T=A), and the subsystem totally failed (P=T) to perform its function. The fault was discovered during a routine test or inspection (D=I) and was consequential to the sequence (EFF=YB). The command fault or failure belongs to the generic class of "subsystem failure" (ICOMP=930).

System

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF	ICOMP
4			RT	EH		XXX					1	A	T	I	YB	920

System-level steps are identifiable by the special use of the component field--the code "XXX" appears. The system step was caused by a previous step(s) (CAUSE=RT). The system involved was the Emergency Power Generation System (PSYS=EH). This step involved the failure of the entire system (COMP=XXX). If the value of 1 for differ (DI=1) appears again in another step involving the total failure of the Emergency Power Generation System, then the two steps describe failures of the same Emergency Power Generation System. It was an actual, immediate fault

(T=A), and the system totally failed (P=T) to perform its function. The fault was discovered during a routine test or inspection (D=I) and was consequential to the sequence (EFF=YB). The command fault or failure belongs to the generic class of "system failure" (ICOMP=920).

Unit Effec'

SCSS also captures information on the effect of the sequence on the entire unit. To capture this information, special codes were developed. A step containing a pseudo system code "XX" in the PSYS field contains information regarding the initial unit conditions and the effects of the sequence on that or other units. Unit data are coded in the T and P fields on an XX step.

The unit information step is usually not part of the sequence, but appears at the end of a sequence. However, it can occur during a sequence to indicate that a previous occurrence(s) caused (1) a reactor trip or shutdown or (2) failures of components or systems in another unit.

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF
5				XX								E	AA		YB

The pseudo PSYS code, XX, indicates a unit effect step. The initial unit condition was steady-state power operation (T=E). The effect of the event on the unit was a manual shutdown (P=AA). Information in this step was consequential to the sequence (EFF=YB).

Environmental Effect

The environmental effect step is similar to the unit effect step. The step is signified by the appearance of "YY" in the PSYS field. YY indicates that the step contains information about radiological releases and personnel exposures resulting from the sequence. Environmental data are coded in the T and P fields on a YY step.

The environmental step, like the unit step, is usually not part of a sequence; however, it can occur during a sequence.

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF
6				YY								N	N		YC

The pseudo PSYS code YY indicates an environmental effect step. No radioactivity was released to the environment (T=N), and no personnel were exposed to radiation (P=N). Information in this step was inconsequential to the sequence (EFF=YC).

FIGURE 4. USE OF CERTAIN FIELDS FOR SIX TYPES OF STEPS

Type of Step	CAUSE	PSYS	COMP	DI	T	P	D	EFF
1. Equipment								
Typical code	EF	CL	HX	1	A	TR	I	HN
Use of field	Actual equipment related cause--electrical, mechanical, etc.	Actual system involved	Actual component involved	Numeric value for equipment steps	Timing of failure--instantaneous, pre-existing, etc.	Equipment performance--total or partial failure, repair required or not	Method of detection--test, maintenance	Effect of step
2. Personnel								
Typical code	SH	PM	PCP	A	M	T	I	TB
Use of field	Special human factor cause codes begin with "S"	Special personnel codes identify type of activity beginning with "P"	Special personnel codes identify type of personnel involved	Alphabetic value for personnel steps	Same as equipment step	Same as equipment step	Same as equipment step	Special personnel codes begin with "T" or "U"
3. Subsystem								
Typical code	BS	EH	1X3	1	A	T	I	YB
Use of field	Same as equipment step	Same as equipment step	Indicates number of trains failed or of number of available trains	Same as equipment step	Same as equipment step	Same as equipment step	Same as equipment step	Same as equipment step
4. System								
Typical code	RT	EH	XXX	1	A	T	I	YB
Use of field	Same as equipment step	Same as equipment step	Special component code to note total system failure	Same as equipment step	Same as equipment step	Same as equipment step	Same as equipment step	Same as equipment step
5. Unit								
Typical code	-	XX	-	-	E	AA	-	YB
Use of field	-	Special psys code for unit effect step	-	-	Indicates initial unit conditions at the start of the event--at power, refueling, etc.	Effect of the event on the unit--automatic scram, power reduction, etc.	-	Indicates if step is consequential or inconsequential to sequence
6. Environmental Effect								
Typical code	-	YY	-	-	N	N	-	YC
Use of field	-	Special psys code for environmental effect step	-	-	Indicates radioactivity released to environment	Indicates exposure to personnel	-	Indicates if step is consequential or inconsequential to sequence

2. TUTORIAL--RETRIEVING, DISPLAYING, AND ANALYZING LER INFORMATION

2. TUTORIAL—RETRIEVING, DISPLAYING, AND ANALYZING LER INFORMATION

2. TUTORIAL - RETRIEVING, DISPLAYING, AND ANALYZING LERs

This chapter is intended to get you started working on the SCSS database and to acquaint you with the log on procedures and the basic commands necessary to retrieve, display, and analyze selected LERs. The examples in this section will give you "hands-on" experience in using many of the fundamental commands and will familiarize you with the strategies and functions employed by the SCSS. More detailed explanations and a full listing of all options available for the system commands are presented in Chaps. 3 and 4.

The appearance of this manual has been standarized as much as possible to make it easier to use. Keep the following 10 points in mind when reading this manual and using the SCSS database:

1. all commands to be entered by the user during the tutorial are displayed in **UPPERCASE BOLDFACE** print;
2. following each command line you must press the RETURN (or ENTER) key;
3. carats <> are used by the SCSS to indicate a field name;
4. parentheses () are used in various combinations to specify the and/or logic used by SCSS; numerous examples illustrating the use of parentheses will be shown and explained in this chapter and in Chap. 3;
5. the SCSS syntax uses blanks as delimiters; therefore, all terms in the command line must be set off from the other terms by blanks;
6. the asterisk (*) indicates that you are in command mode;
7. the plus (+) prompt indicates that the computer is requesting more information;
8. the END command signifies that you have finished a command sequence and tells the computer to execute the operation using the criteria you have specified;
9. if you have made a mistake on a multi-line command, you can enter ABORT

on a separate line and the computer will return you to the command entry level (*) without executing any commands; and

10. if you make a typographical error when you are entering commands, you can correct the mistake by striking the underline key the same number of times as the number of incorrectly typed letters. For example, incorrectly typing the HELP command as HELOP would be corrected by striking the underline key twice (one for the O and one for the P) and then retyping the P (i.e., HELOP _ _ P).

LOGGING ON TO THE SYSTEM

Two telephone lines (300 baud and 1200 baud) are available for accessing the SCSS database, which resides on the ORNL IBM-3033 computers. Before you dial the system, be sure your terminal is configured with the following parameters (ask the computer operators or the support staff for assistance to ensure that these are met):

- Baud rate - 300 or 1200
- Duplex - Full
- Parity - Odd
- Parity Sense - Off

If your terminal is set at 300 baud, follow the instructions in Table 2-1 to log on to the SCSS database. If your terminal is set at 1200 baud, follow the instructions in Table 2-2 to log on to the SCSS database. After you have logged on to the system, continue on p. 2-5, "Retrieving LER Records."

In some cases, a dropped phone line or other inadvertent action by a user will result in the loss of the connection with the host computer. If this should happen, log back on to the computer using the same logon procedures shown in Tables 2-1 and 2-2. When the computer requests your logon ID, enter your three-character logon initials followed by a space and the word "RECONNECT" and then continue with the rest of the logon procedure.

TABLE 2-1. LOGON PROCEDURES FOR A 300-BAUD CONNECTION

1. Dial **(615) 574-7620** or **FTS 624-7620.**

2. After establishing communications with the computer, type

   ATLOGON in upper case and depress the **RETURN** key.

NOTE: Do not use lower case and do not depress the return key before typing ATLOGON because these actions will result in a lost connection.

The computer system will respond with the message:

```
ENTER LOGON ID
```

3. Type in your logon identification **'Your ID'** then depress the **RETURN** key.

The system will respond with a request for your password:

```
ENTER PASSWORD
XXXXXX
```

4. Type in your password **'Password'** then depress the **RETURN** key.

The computer will provide general system messages and the following:

```
DO YOU WISH TO SEE THE CURRENT SCSS SYSTEM MESSAGES? (YES/NO)
```

5. Type in **YES** or **NO** then depress the **RETURN** key.

After this step, the system will respond with the message:

```
THIS SESSION IS READY TO BEGIN
*
```

The asterisk signifies that the system is ready for commands.

TABLE 2-2. LOGON PROCEDURES FOR A 1200-BAUD CONNECTION

1. Dial **(615) 574-7474** or **(615) 576-5750**
   or **FTS 624-7474** or **FTS 624-7620**

2. After establishing contact with the computer,
   depress the **RETURN** key twice.

   The computer system will respond with the message:

   ```
   CENTRAL NETWORK HOST
   WHICH SYSTEM (1 THRU 50) ?
   ```

3. Type **F2** then depress the **RETURN** key twice.

   The system will respond with the following message:

   ```
   SELECT A SYSTEM
   TYPE RECON, TLOGON, I FOR INTERACT, OR HELP
   ```

4. Type **TLOGON** then depress the **RETURN** key.

   The system will respond with the following message:

   ```
   ENTER LOGON ID
   ```

5. Type in your logon identification **'Your ID'** then depress the **RETURN** key.

   The system will respond with a request for your password:

   ```
   ENTER PASSWORD
   XXXXXX
   ```

6. Type in your password **'Password'** then depress the **RETURN** key.

   The computer will provide general system messages and the following:

   ```
   DO YOU WISH TO SEE THE CURRENT SCSS SYSTEM MESSAGES? (YES/NO)
   ```

7. Type in **YES** or **NO** then depress the **RETURN** key.

   After this step, the system will respond with the message:

   ```
   THIS SESSION IS READY TO BEGIN
   *
   ```

   The asterisk signifies that the system is ready for commands.

RETRIEVING LER RECORDS

Now that you have logged on to the SCSS database, you can begin to enter commands to retrieve specific groups of LER records. Each command discussed in this section is introduced by a short explanation of its function, a description of its use in the particular search strategy, the commands you will use to carry out the search, the resulting computer output (highlighted as a simulated computer screen), and finally an interpretation of the output produced by the command. This discussion begins with the most frequently used SCSS command, FIND HELP.

The FIND HELP command, a variation of the FIND command, allows you to conduct a search of the SCSS database using a textual (rather than coded) variable description. In the first example we are interested in obtaining information on personnel actions contained in the LERs, but do not know the codes that are used by the system to access them. Therefore, we use the FIND HELP command as the entry point, both to do our initial search and to obtain the SCSS codes. The first commands entered below will be used to retrieve all the LERs that contain the text "personnel" in the decoded component (COMP) field. To conduct the search, follow these directions:

1. At the asterisk (*); type

   FIND HELP

2. The computer will respond with a plus (+) prompt, which means it is ready for more specific information; type

   <COMP> PERSONNEL

   The screen will show personnel codes from the component field and a second + prompt.

```
SYMBOL:PCP    CONTRACTOR PERSONNEL
SYMBOL:PLO    LICENSED OPERATOR PERSONNEL
SYMBOL:PNA    PENETRATION, PERSONNEL ACCESS
SYMBOL:PNO    NON-LICENSED OPERATOR PERSONNEL
SYMBOL:PUX    OTHER/ UNKNOWN UTILITY PERSONNEL
SYMBOL:PX     OTHER PERSONNEL
SYMBOL:PZ     UNKNOWN PERSONNEL
+
```

3. To end the command sequence and begin the search; type

END

The computer will respond with the number of LERs retrieved.

==>GROUP 1 HAS 5273 LERs--OPERATION: "FIND"

As you can see, this search has retrieved 5237 LERs that all contain the decoded text "personnel" in the component field; it has also listed the SCSS codes for components (PCP, PLO, PNA, PNO, PUX, PX, and PZ) that can be used in subsequent searches. Group 1, the most recently formed collection of LERs, is now considered to be the "active record group." As each new record group is formed, it becomes the active record group and receives a sequential number. Note that each time the FIND command is used, the group number is reinitialized to 1. The following illustrates how these commands should have appeared on your screen.

```
*
FIND HELP
+
<COMP> PERSONNEL
SYMBOL:PCP    CONTRACTOR PERSONNEL
SYMBOL:PLO    LICENSED OPERATOR PERSONNEL
SYMBOL:PNA    PENETRATION, PERSONNEL ACCESS
SYMBOL:PNO    NON-LICENSED OPERATOR PERSONNEL
SYMBOL:PUX    OTHER/ UNKNOWN UTILITY PERSONNEL
SYMBOL:PX     OTHER PERSONNEL
SYMBOL:PZ     UNKNOWN PERSONNEL
+
END
 ==>GROUP 1 HAS 5273 LERs--OPERATION  "FIND"
```

The next example illustrates a slightly more complex search in which we are interested in retrieving LERs that involve the personnel actions of omitting a task or step. This example will also illustrate how particular codes can be deleted or selected from the total set of codes available. Again, because we do not know which codes to use, we will use the FIND HELP command.

1. At the asterisk (*); type

FIND HELP

2. When the computer responds with a plus (+) prompt; type

<COMP> PERSONNEL

The computer will respond as it did in the last example, giving the personnel codes from the component field and another + prompt.

```
SYMBOL:PCP    CONTRACTOR PERSONNEL
SYMBOL:PLO    LICENSED OPERATOR PERSONNEL
SYMBOL:PNA    PENETRATION, PERSONNEL ACCESS
SYMBOL:PNO    NON-LICENSED OPERATOR PERSONNEL
SYMBOL:PUX    OTHER/ UNKNOWN UTILITY PERSONNEL
SYMBOL:PX     OTHER PERSONNEL
SYMBOL:PZ     UNKNOWN PERSONNEL
+
```

3. To delete a particular code from the active group (e.g., PNA), type

DELETE PNA
+

4. To extend the search to include coded values for the effect field that contain "omission" in the description, type

<EFF> OMISSION
+

The computer will respond with the codes that contain omission in the effect field and another + prompt.

```
SYMBOL:TA    INCONSEQUENTIAL OMISSION OF TASK, ANALYSIS OR STEP
SYMBOL:TB    CONSEQUENTIAL OMISSION OF TASK, ANALYSIS, OR STEP
SYMBOL:TX    OTHER OMISSION HUMAN FACTOR
SYMBOL:TC    CONSEQUENTIAL OMISSION WITHIN ALLOTED TIME
SYMBOL:TD    CONSEQUENTIAL OMISSION OF ALARM RESPONSE
SYMBOL:TE    CONSEQUENTIAL OMISSION OF ADJUSTMENT OR CALIBRATION
+
```

5. To search for the remaining six personnel codes and only the TB effect code (CONSEQUENTIAL OMISSION OF TASK, ANALYSIS, OR STEP), type

SELECT TB
+

6. To end the command sequence and begin the search, type

END

The computer will respond with the number of LERs retrieved.

```
==>GROUP 1 HAS  1093 LERs--OPERATION: "FIND"
```

This second use of the FIND HELP command illustrates the use of two search strategies as well as the use of the DELETE and SELECT options. The decoded fields for component and effect were searched for the values personnel and omission, respectively. After the LERs that contained these values were retrieved, we deleted the PNA code because PENETRATION, PERSONNEL ACCESS was not an actual type of personnel. We retained the other six component codes and selected the TB code as the value to be searched on in the effect field. This search then selected all LERs that contained the decoded text "personnel" (except code PNA) in the component field **and** TB in the effect field. The following illustrates how the sequence of commands should have appeared on your screen.

```
*
FIND HELP
+
<COMP> PERSONNEL
SYMBOL:PCP    CONTRACTOR PERSONNEL
SYMBOL:PLO    LICENSED OPERATOR PERSONNEL
SYMBOL:PNA    PENETRATION, PERSONNEL ACCESS
SYMBOL:PNO    NON-LICENSED OPERATOR PERSONNEL
SYMBOL:PUX    OTHER/ UNKNOWN UTILITY PERSONNEL
SYMBOL:PX     OTHER PERSONNEL
SYMBOL:PZ     UNKNOWN PERSONNEL
+
DELETE PNA
+
<EFF> OMISSION
SYMBOL:TA    INCONSEQUENTIAL OMISSION OF TASK, ANALYSIS OR STEP
SYMBOL:TB    CONSEQUENTIAL OMISSION OF TASK, ANALYSIS, OR STEP
SYMBOL:TX    OTHER OMISSION HUMAN FACTOR
SYMBOL:TC    CONSEQUENTIAL OMISSION WITHIN ALLOTED TIME
SYMBOL:TD    CONSEQUENTIAL OMISSION OF ALARM RESPONSE
SYMBOL:TE    CONSEQUENTIAL OMISSION OF ADJUSTMENT OR CALIBRATION
+
SELECT TB
+
END
        ==>GROUP 1 HAS  1093 LERs--OPERATION: "FIND"
```

The next examples will illustrate how using parentheses with field values will allow you to conduct more complicated searches using and/or Boolean logic.

Using the codes we have just obtained from the FIND HELP searches, we will now use the FIND command to search for particular instances of selected component personnel codes and for particular effect codes. We will start this new search by using the FIND command.

1. At the asterisk (*); type

   FIND

2. The computer will respond with the + prompt; type

   <COMP> (PLO PNO PUX)

3. The computer will respond with the + prompt; type

   END

   The computer will respond with the following message:

   ==>GROUP 1 HAS 3458 LERs--OPERATION "FIND"

This search has retrieved 3458 LERs, each containing the code PLO **or** PNO **or** PUX in the component field. The parentheses around all three codes in this search indicate that the different component codes are searched for using the **"or"** Boolean logic. The following illustrates how this search should appear on your screen.

```
*
FIND
+
<COMP> (PLO PNO PUX)
+
END
==>GROUP  1 HAS  3458 LERs--OPERATION: "FIND"
```

The next search, a modification of the previous one, will look for LERs that contain a particular combination of two of the three component codes (PLO, PNO, PUX). However, instead of looking for any occurrence of these codes, we will now look for LERs that satisfy either of two conditions: they contain the component codes PLO and PUX **or** they contain the component codes PNO and PUX. Enter the following commands.

1. At the asterisk (*); type

   FIND

2. The computer will respond with a + prompt for more information; type

   <COMP> (PLO PNO) PUX

3. The computer will respond with a + prompt; type

   END

   The computer will display the results of the search.

   ==>GROUP 1 HAS 94 LERs--OPERATION "FIND"

As you can see, the conditions specified on this search have narrowed the number of LERs in the active group to 94. In this example, the parentheses around the first two codes and not the third denote that the first two meet the "or" condition, whereas the condition for the third code is "and" with either of the first two in parentheses. The following illustrates how the sequence of commands should appear on your screen.

```
*
FIND
+
<COMP> (PLO PNO) PUX
+
END

==>GROUP 1 HAS    94 LERs--OPERATION: "FIND"
```

The next example will illustrate the search strategy used for locating LERs that contain all three component codes, PLO **and** PNO **and** PUX.

1. At the asterisk (*); type

   FIND

2. The computer will respond with the + prompt; type

   <COMP> PLO PNO PUX

3. Again, the computer will respond with a + prompt; type

END

The computer will respond with the results of the search.

==>GROUP 1 HAS 3 LERs--OPERATION: "FIND"

The absence of any parentheses around the codes denotes the "and" condition among all three values. As you can see, the search for LERs that contain all three component codes, PLO **and** PNO **and** PUX, resulted in only three LERs being retrieved. The following illustrates how the search sequence should appear on your screen.

```
*
FIND
+
<COMP> PLO PNO PUX
+
END

==>GROUP 1 CONTAINS     3 LERs--OPERATION: "FIND"
```

In summary, these examples have shown that field values contained within parentheses denote the "or" condition, whereas field values outside of parentheses denote the "and" condition.

The SCSS database can also be searched for coded values within multiple fields (e.g., component codes and effect codes). In this next example, a search will be conducted for either of two personnel codes already used, PLO and PNO, **and** for an effect code, TB. Two searches will be conducted to retrieve any LER records that contain all three of these types of information and to illustrate the use of different fields in the FIND command.

1. At the asterisk (*); type

FIND

2. The computer will respond with the + prompt for more information; type

<COMP> (PLO PNO) <EFF> TB

3. The computer will respond with the + prompt; type

END

The computer will display the results of the search.

==>GROUP 1 HAS 285 LERs--OPERATION: "FIND"

This search looks for records that have either of two combinations of the component and effect code values. The parentheses around the two component values denote the "or" condition for the component field; however, because effect TB is not included in the parenthenses, the effect field must occur with either of these two codes ("and" condition). The search was conducted, then, for either of two conditions; PLO and TB **or** PNO and TB. This search resulted in 285 LERs that meet these conditions. The following screen illustrates the appearance of this command sequence.

```
*
FIND
+
<COMP> (PLO PNO) <EFF> TB
+
END
==>GROUP  1 HAS    285 LERs--OPERATION: "FIND"
```

Finally, we will conduct a search to retrieve the LERs that contain all three codes from both the component and the effect fields searched for previously.

1. At the asterisk (*); type

FIND

2. The computer will respond with the + prompt for more information; type

<COMP> PLO PNO <EFF> TB

3. The computer will respond with the + prompt; type

END

The computer will display the results of the search.

==>GROUP 1 HAS 6 LERs--OPERATION: "FIND"

This search illustrates a strategy that looks for records that have all three values from the component and the effect fields. All records that contain the codes PLO **and** PNO **and** TB in these two fields have been retrieved. You can see that as the conditions of the search become more stringent, fewer records are retrieved. This search resulted in an active record group of only six LERs. The following illustrates how this sequence should appear on your screen.

```
*
FIND
+
<COMP> PLO PNO <EFF> TB
+
END

==>GROUP  1 HAS      6 LERs--OPERATION: "FIND"
```

For the remainder of the tutorial, simulated computer screens will be displayed to illustrate the commands to be entered. Remember:

1. all the commands you should enter are displayed on the simulated screens in **UPPERCASE BOLDFACE** print;
2. following each command line, you must hit the RETURN key;
3. the asterisk (*) prompt indicates that you are in command mode;
4. the plus (+) prompt indicates that the computer is requesting more information;
5. the END command signifies that you have finished a command sequence; and
6. to correct typographical errors, strike the underline key the same number of times as the number of incorrectly typed letters.

Before proceeding with further LER retrieval examples, we will briefly discuss the various types of help commands available. These help commands allow you to obtain online assistance for searching techniques, to obtain code values for information desired to be included in the search logic, and to list information on syntax and use for the commands in the system.

The command HELP first provides a list of system commands that are available, then can provide information on their use. Start by typing **HELP** on your terminal at the asterisk (*) prompt.

The simulated screen in Fig. 2-1 shows the output from the HELP command, a list of all SCSS commands. To obtain information on any one of the commands listed, type HELP and the command name. The detailed information output from the HELP command includes the correct syntax, usage, and examples. For example, if you wished to obtain information about the LIST command, use the following command.

```
*
HELP LIST

The LIST command is used to display LERs.  Several variations in output
format can be selected using the OPTIONS command.

Allowed Abbreviation:  LI or L

Syntax:  LIST
         (to list the first LER in the currently active record group)

    OR:  LIST ddd yy nnn
         (to list the LER that is identified by docket number ddd,
         in year yy, and LER number nnn)

    or:  LIST n
         (to list the nth LER in the currently active record group)

    or:  LIST n1 n2
         (to list LERs in positions n1 through n2 from the currently
         active record group.  In the case that more than 10 LERs are
         requested, a "YES" response must be entered to a prompting
         the program issues, to ensure that the user has not
         inadvertently requested the listing of more forms than were
         wanted.)
```

Another help command is the AUTH command, which allows you to search a file that contains descriptive text for each of the codes used in the database. This authority file is used to obtain key codes for words or text strings in one or more of the 11 coded fields. The authority file is simply a directory of acceptable codes and their corresponding text strings. The AUTH command does not retrieve LERs, but it does identify the codes you need if you wish to obtain information on certain fields or field values. Entering the text to be searched allows you to view the different codes associated with that text. Type the following command.

FIGURE 2-1. OUTPUT FROM THE HELP COMMAND

```
                    SCSS COMMAND CHART

1. Locate the Initial Group of LERs in a Sequence of Searching
   Operations.
      FIND, XFIND, BUILD, ACTIVE

2. Screen a Group of LERs.
      SCAN, STEPSCAN, LINK, JOIN, TEXTSCAN, LOCATE

3. Process a Group of LERs.
      VALUES, RELATE, TREND, REF

4. Process "Captured" Step Data Collections.
      STEPLIST, SVALUES, SRELATE, STEPSORT

5. List Record Keys and LERs.
      LIST, OPTIONS, KEYSORT, DISPLAY

6. Manipulate and Combine Record Groups.
      COMBINE, MATCH, NOMATCH, NOTFOUND, RECALL, REMOVE

7. Display Searching Logic.
      TABLE

8. Request Assistance.
      HELP, DOCKET, INVERTED, AUTH

9. Change the Record Searching Hierarchy.
      DEFINE, DEFAULT

10. Manipulate Saved Searching Strategies.
      ACTIVE, SAVE, DELETE, EXEC, DIRECTORY

11. Terminate this Session.
      STOP

Enter HELP HELP for more information on the commands or enter
      HELP Command
for detailed information on an individual command.
Additional aid can be obtained by typing:
   HELP SYNTAX -- describes syntax for searching commands
   HELP FIELDS -- lists available fields
   HELP FILES  -- lists files in the SCSS System
```

```
AUTH <COMP> PERSONNEL


FIELD:COMP  VAR:PCP  --CONTRACTOR PERSONNEL
FIELD:COMP  VAR:PLO  --LICENSED OPERATOR PERSONNEL
FIELD:COMP  VAR:PNA  --PENETRATION, PERSONNEL ACCESS
FIELD:COMP  VAR:PNO  --NON-LICENSED OPERATOR PERSONNEL
FIELD:COMP  VAR:PUX  --OTHER/ UNKNOWN PERSONNEL
FIELD:COMP  VAR:PX   --OTHER PERSONNEL
FIELD:COMP  VAR:PZ   --UNKNOWN PERSONNEL
```

As you can see, the codes associated with the text "personnel" in the component field include PCP, PLO, PNA, PNO, PUX, PX, and PZ. These codes can now be used to formulate and construct searching strategies if you are interested in any type of personnel coded as a component. As you may recall, these codes were used in the first examples of the tutorial.

The DOCKET command is another type of help command used to search for specific nuclear power plant information contained in the docket information section of the LER. The searchable data fields include the docket number, the name of the facility, reactor type, NRC region, NSSS supplier, architectural engineer, and facility operator. For example, to find information on Three Mile Island, type the following command.

```
*
DOCKET THREE MILE ISLAND
DOCKET:289  THREE MILE ISLAND 1      TYPE:PWR
              REGION:  1             NSSS:B&W
ARCHITECTURAL ENGINEER: GIL
     FACILITY OPERATOR: METROPOLITAN EDISON CO.
              SYMBOL: MEC

DOCKET:320  THREE MILE ISLAND 2      TYPE:PWR
              REGION:  1             NSSS:B&W
ARCHITECTURAL ENGINEER: BURNS/ROE
     FACILITY OPERATOR: METROPOLITAN EDISON CO.
              SYMBOL: MEC
```

The two power plant units listed above are the docket records for Three Mile Island. Like the authority file, the docket file is a part of the SCSS database that contains information for use in your searching logic, but does not contain information on any specific LERs. For example, if you were interested in finding information on all nuclear power plants in Region 5 that use a BWR-type reactor, you would type the following:

```
*
DOCKET TYPE=BWR REGION=5

DOCKET:133  HUMBOLDT BAY                TYPE:BWR
              REGION:  5                NSSS:GE
ARCHITECTURAL ENGINEER: BECH
     FACILITY OPERATOR: PACIFIC GAS & ELECTRIC CO.
              SYMBOL: PGE

DOCKET:397  WPPSS 2                     TYPE:BWR
              REGION:  5                NSSS:GE
ARCHITECTURAL ENGINEER: BURNS/ROE
     FACILITY OPERATOR: WASH. PUBLIC POWER SUPPLY SYSTEM
              SYMBOL: WPP

DOCKET:522  SKAGIT 1                    TYPE:BWR
              REGION:  5                NSSS:GE
ARCHITECTURAL ENGINEER: BECH
    FACILITY OPERATOR: PUGET SOUND POWER & LIGHT CO.
              SYMBOL: PUG

DOCKET:523  SKAGIT 2                    TYPE:BWR
              REGION:  5                NSSS:GE
ARCHITECTURAL ENGINEER: BECH
    FACILITY OPERATOR: PUGET SOUND POWER & LIGHT CO.
              SYMBOL: PUG
```

As these examples show, the DOCKET command can be used to obtain specific nuclear power plant information. The information obtained via the DOCKET command can then be used as part of your searching logic to retrieve specific LERs using the LOCATE command.

The LOCATE command scans the LERs in the active record group and examines the information contained in the docket file. As discussed earlier, this information includes the docket code, the name of the facility, the plant type, the NRC region, the NSSS supplier, the architectural engineer, and the facility operator code. The LOCATE command searches for information in these fields and selects the LER records that satisfy your specifications. This

command helps to narrow your search when you are interested in only locating LERs based on information in the docket file.

In this next example, we will be interested in LERs that contain the component code PUX with either PLO or PNO. Once we have obtained these LERs using the FIND command, we will narrow the retrieval (using the LOCATE command) to include only those LERs that involve pressurized water reactors (PWR). To obtain this information, type in the following sequence of commands.

```
*
FIND
+
<COMP> (PLO PNO) PUX
+
END

==>GROUP 1 HAS    94 LERs--OPERATION: "FIND"
*
LOCATE TYPE=PWR
LERS WILL BE KEPT FOR PWR  REACTORS
==>GROUP  2 HAS     59 LERs OUT OF    94 SEARCHED--OPERATION: "LOCATE"
```

The group resulting from this search contains 59 LERs, each of which has a plant type coded as PWR.

The CHART option of the LOCATE command allows you to analyze LERs that were retrieved by the FIND and LOCATE commands. This option produces a chart of the names of the plants, the docket number, and the year that the plant submitted the record. To obtain this information on the current record group of 59 LERs, enter the following command.

LOCATE CHART

==GROUP 3 HAS 59 LERs OUT OF 59 SEARCHED--OPERATION: "LOCATE"

COUNT ON LERS SUBMITTED

FACILITY	DOCKET	1980	1981	1982	1983	1984	TOTAL
YANKEE ROWE	29	0	0	0	1	0	1
CONNECTICUT YANKEE	213	0	1	0	0	0	1
GINNA	244	0	0	0	1	0	1
TURKEY POINT 4	251	0	0	1	0	0	1
PALISADES	255	0	0	0	0	1	1
POINT BEACH 1	266	0	1	0	0	0	1
SALEM 1	272	0	1	0	0	0	1
DIABLO CANYON 1	275	0	0	0	1	0	1
SURRY 2	281	0	0	1	0	0	1
ZION 1	295	0	0	1	0	1	2
SALEM 2	311	0	1	2	1	0	4
RANCHO SECO	312	0	0	0	1	2	3
COOK 1	315	0	0	1	1	0	2
COOK 2	316	0	1	0	0	0	1
CALVERT CLIFFS 1	317	0	1	1	1	0	3
THREE MILE ISLAND 2	320	0	1	0	1	0	2
SEQUOYAH 1	327	0	0	1	0	0	1
SEQUOYAH 2	328	0	2	0	1	1	4
BEAVER VALLEY 1	334	0	1	1	0	0	2
ST. LUCIE 1	335	0	0	1	0	0	1
NORTH ANNA 1	338	0	1	1	0	0	2
NORTH ANNA 2	339	0	2	0	0	0	2
TROJAN	344	0	0	0	1	0	1
DAVIS-BESSE 1	346	0	2	1	1	0	4
FARLEY 1	348	0	0	0	0	1	1
SAN ONOFRE 2	361	0	0	1	0	0	1
SAN ONOFRE 3	362	0	0	0	2	0	2
MCGUIRE 1	369	0	2	3	1	0	6
MCGUIRE 2	370	0	0	0	2	1	3
ST. LUCIE 2	389	0	0	0	2	0	2
SUMMER 1	395	0	0	0	1	0	1
		0	17	16	19	7	59

As shown, the LOCATE CHART option allows you to analyze the number of submissions by each plant over time. In this search, the relative number of plant submissions was consistent for years 1981, 1982, and 1983. LERs for 1980 are 0 because they have not, as yet, been added to the SCSS database; LERs for 1984 include only those submitted through May of 1984.

TEXTSCAN, like LOCATE, is another SCSS command that examines LERs initially retrieved with the FIND command. TEXTSCAN searches each of the abstracts

in the active record group of LERs and looks for specific text strings that you have selected. You can search for one or two text strings of up to 64 characters each using one command. Note, however, that only one text string per line is allowed. Also, each text string must be set off with single quotation marks, as shown in the example. System options that allow you to specify the order of the searches are discussed in Chap. 3.

For this example, we will examine the current group of LERs to determine if any of the abstracts contain the text strings 'reactor' and 'test'.

```
*
TEXTSCAN
+
'REACTOR'
+
'TEST'
+
END
==>GROUP  4 HAS     29 LERs OUT OF    59 SEARCHED--OPERATION: "ABSTRACT"
```

As you can see, the TEXTSCAN command retrieved 29 LERs from the active record group of 59. All of the 29 LERs retrieved contain the text strings 'reactor' or 'test' in their abstracts.

Thus far our example searches have been conducted on the entire LER record, either in the full coded step matrix to look for specific occurrences of component or effect codes (FIND), or in the docket file (LOCATE), or in the abstract to look for specific text strings (TEXTSCAN). Another SCSS command, STEPSCAN, follows the same type of logic as the other searches, but instead of looking at the entire LER record, it looks for information in an individual occurrence step (NOTE: You may wish to refresh your memory on the structure of the SCSS database by referring to Chap. 1 for specific details.) For instance, when we searched for LERs that contained the component code values PLO and PNO using the FIND command, we were looking for LERs that contained those values on any lines in the coded step matrix. Now we want to conduct searches for LERs that contain specific codes on the same step (line) of the coded step matrix, so we will use the STEPSCAN command. Because the STEPSCAN command looks only at individual steps within the step matrix, each field can

only have one value; searching for multiple values within a single field would be meaningless. STEPSCAN, then, is used to locate LERs that have specified values in any number of different fields within the same step. All fields within the step matrix can be searched with the STEPSCAN command.

To illustrate the STEPSCAN command, we will retrieve those LERs from the active record group (group 4) that contain the effect value TB in the same step as the component value PLO.

```
*
STEPSCAN
+
<EFF> TB <COMP> PLO
+
END
THE FOLLOWING "STEP" DATA COLLECTIONS WILL BE CAPTURED:
    THE STEPS SATISFYING THE CURRENT SEARCHING LOGIC
==>GROUP  5 HAS     11 LERs OUT OF     29 SEARCHED--OPERATION: "STEPSCAN"
     12 STEPS SATISFIED THE SEARCH STRATEGY
               SECONDS:      0.11 (CPU)      3.46 (CLOCK)--RATIO:0.032
```

This search indicates that 11 LERs in the active group meet the condition of the search; that is, 11 LERs out of 29 have records with a component code equal to PLO and a effect code equal to TB in the same step. Also note that 12 steps satisfied the search strategy, indicating that one of the LERs contained two steps that had a PLO component value and a TB effect value in the same step.

Now let's review the retrieval steps that we have conducted thus far. The 11 LERs in the active record group (group 5) have all met a number of conditions: (1) they have either of two component code combinations (PLO with PUX) or (PNO with PUX) codes; (2) they have pressurized water reactors; (3) their abstracts contain the text strings 'reactor' or 'test'; and (4) they contain both the component code PLO and the effect code TB in the same step.

As you can see from the above example, the logic used in retrieving LERs can sometimes become complicated. The TABLE command allows you to decode the searching logic employed in the different searching commands. TABLE interprets the commands entered and produces a table that lists the logic you

have specified in your searches. If, for instance, we conduct a search for a combination of component codes and for the effect code from the earlier searches, we can retrieve another group of LERs and check our searching strategy. The simulated screen below shows the commands you should enter and how they will appear on your screen.

```
*
FIND
+
<COMP> (PNO PLO) PUX <EFF> TB
+
END
==>GROUP 1 HAS        46 LERs--OPERATION:"FIND"
*
TABLE FIND
----------------------"FIND"----------------------
     COMP =PNO --NON-LICENSED OPERATOR PERSONNEL
       OR =PLO --LICENSED OPERATOR PERSONNEL
 AND COMP =PUX --OTHER/ UNKNOWN UTILITY PERSONNEL
 AND EFF  =TB  --CONSEQUENTIAL OMISSION OF TASK, ANALYSIS, OR STEP
```

TABLE allows you to check your searching logic for the FIND command and shows that you did, indeed, capture the LERs you had been searching for, those that have the effect value TB and the component values PNO and PUX or PLO and PUX.

DISPLAYING LER RECORDS

Up to this point we have been illustrating various methods for retrieving specific LER records from the SCSS database. Once the LERs have been retrieved, there are a number of ways to display the information they contain. The DISPLAY, LIST, and OPTIONS LIST commands help you to display the information from the collected group of LERs. The DISPLAY command lists the docket numbers, year of the event, and LER numbers for all LERs in the active record group; the LIST and OPTIONS LIST commands display the partial or complete contents of an LER record.

In this next example we will first retrieve all LER records that contain PLO, PNO, and PUX in the component field and TB in the effect field. After these

records have been retrieved, we will display and list them. The simulated screen illustrates the commands to be entered and the resulting responses.

```
*
FIND
+
<COMP> PLO PNO PUI  <KFF> TB
+
END

==>GROUP 1 HAS     2 LERs--OPERATION: "FIND"
*
DISPLAY
    1 325/81-093     2 389/83-052
```

As you can see, only two LERs were retrieved with the FIND command. The DISPLAY command has listed the two LER numbers (325/81-093 and 389/83-052) with their accompanying identification numbers (1 and 2) from the record group.

The next command discussed, LIST, is used to provide a listing of the LER record(s). This listing can display either the entire record or only portions of the record using the OPTIONS command (presented next). The LIST command will list any LER in the active record group, and the record to be listed can be specified using either (1) the LER number or (2) the identifying number from the record group (for instance, when we displayed the two LERs above, the identifying numbers were 1 and 2). Either method of selecting the record will result in identical listings. Note, however, that when you are using the docket identification number to list the record, you must enter the digits as in the record but without the slash (/) or the hyphen (-) and with blanks where these characters are. The simulated screen in Fig. 2-2 illustrates the use of the command and the output produced by the LIST command.

The OPTIONS LIST command is similar to the LIST command except that it allows you to select only that information you want to be listed without having to print out the entire record. Note that changing the information to be listed and then listing it is a two-step process: first, you specify the options you want invoked (using the OPTIONS LIST command), and second you enter the LER numbers or the identification numbers you want listed (using

FIGURE 2-2. EXAMPLE OF LIST

```
*
LIST 325 81 093

                              LER SCSS DATA                              08-02-84
**********************************************************************************

DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER    NSIC      EVENT DATE
  325   1981      093         2     8206240192  174224     12-26-1981

DOCKET:325  BRUNSWICK 1                  TYPE:BWR
                 REGION:  2              NSSS:GE
ARCHITECTURAL ENGINEER: UE&C
     FACILITY OPERATOR: CAROLINA POWER & LIGHT CO.
                 SYMBOL: AHV

ABSTRACT
 ON DECEMBER 31, 1981, DURING A DISCUSSION OF PROBLEMS ASSOCIATED WITH
 NON-TECHNICAL SPECIFICATION RELATED INSTRUMENTATION, IT WAS BROUGHT TO
 THE DUTY SRO'S ATTENTION THAT THE RPS VESSEL LOW LEVEL TRIP
 INSTRUMENT, 1-B21-LT-N017D-1, WAS INDICATING UPSCALE.  A REVEIW OF
 PLANT DOCUMENTATION REVEALED THIS PROBLEM WAS FIRST IDENTIFIED ON
 AUXILIARY LOGS ON DECEMBER 26, 1981; HOWEVER, THE APPROPRIATE ACTION
 STATEMENT WAS NOT ENTERED.  THIS COULD HAVE CAUSED A FAILURE TO SCRAM
 AT 162.5 INCHES IF ADDITIONAL CHANNELS WOULD ALSO FAIL. SEE:  TECH
 SPECS 3.3.1A, 6.9.1.8B.  THIS EVENT OCCURRED BECAUSE OPERATIONS
 PERSONNEL FAILED TO RECOGNIZE AND PERFORM THE TECH SPEC REQUIRED
 ACTION WITHIN THE SPECIFIED TIME FRAME.  THE APPROPRIATE RPS ACTION
 STATEMENT WAS IMMEDIATELY ENTERED.  INVOLVED PERSONNEL HAVE BEEN
 COUNSELED ON THE IMPORTANCE OF PROMPT AND THOROUGH REVIEW OF
 IDENTIFIED INSTRUMENT PROBLEMS.
```

the LIST command). For this example, we want to list the matrix (MATRIX) and the decoded step matrix (DECODE) but not the comments (NOCOMM), reference LERs (NOREF), watch-list codes (NOWATCH), or abstract (NOABST). Figure 2-3 illustrates the commands and the resulting output from the OPTIONS LIST and LIST commands.

ANALYZING LER RECORDS

A number of commands allow you to form groups of LERs for analysis by combining records from different searches. This process can proceed by combining two groups of LERs with the COMBINE command or matching identical members from two different collections of LERs with the MATCH command. Group designations of LERs can be retained with the XFIND command and then returned to the active record group with the RECALL command. Groups of LERs that have been combined or matched can then be re-sorted using the KEYSORT command.

The following examples will demonstrate how these commands can be used together to create different groupings of LERs for analysis.

```
*
FIND
+
<COMP> PLO PNO <EFF> TB
+
END
==> LER GROUP 1 HAS 6 LERS -- OPERATION: "FIND"
*
LOCATE TYPE=PWR
==> LER GROUP 2 HAS 2 LERS -- OPERATION: "LOCATE"
*
XFIND
+
<COMP> PLO PUX <EFF> TB
+
END

==>LER GROUP 3 HAS    40 LERs--OPERATION: "XFIND"
```

FIGURE 2-3. EXAMPLE OF OPTIONS LIST

```
*
OPTIONS LIST NOCOMM NOREF NOWATCH NOABST DECODE  MATRIX
CURRENT OPTIONS FOR LIST     COMMAND
  DOCKET
  NOCOMM
  MATRIX
  NOWATCH
  NOREF
  DECODE
  NOABST
  NOTPD
  NOFLOW
*
LIST 1
FORM     1                LER SCSS DATA                          08-02-84
**************************************************************************

DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER   NSIC      EVENT DATE
  325   1981       093          2  8206240192  174224     12-26-1981

DOCKET:325  BRUNSWICK 1              TYPE:BWR
                 REGION:  2          NSSS:GE
ARCHITECTURAL ENGINEER: UE&C
     FACILITY OPERATOR: CAROLINA POWER & LIGHT CO.
                 SYMBOL: AHV

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0   A   ZZ   IU   AD  LT   R369  1    Z  1  1 M T  R  LF
  2   1   A   SC   PO       PNO        1          A M T  R  TC
  3   1   A   SC   PO       PLO        1          B M T  R  TC
  4   1   A   SC   PA       PUX        1          C M T  R  TB
  5   A       RS   IU   AD  1XZ        1    Z  1  1 M T  R  YB
  6             XX                                 E XX     YC
  7             YY                                 N N      YC

------STEP:  1
          SUBLINK:A
            CAUSE:ZZ  --UNKNOWN
   PRIMARY SYSTEM:IU  --REACTOR PROTECTION
 SECONDARY SYSTEM:AD  --REACTOR VESSEL
        COMPONENT:LT  --TRANSMITTER, LEVEL
           VENDOR:R369--ROSEMOUNT, INC.
            TRAIN:Z   --UNKNOWN
           EFFECT:LF  --ERRONEOUS SIGNAL
```

FIGURE 2-3, Continued

```
------STEP:  2
  THIS STEP IS DIRECTLY LINKED TO STEP  1
          SUBLINK:A
            CAUSE:SC  --UNAWARE OF REQUIREMENT
   PRIMARY SYSTEM:PO  --OPERATION ACTIVITY
        COMPONENT:PNO --NON-LICENSED OPERATOR PERSONNEL
           EFFECT:TC  --CONSEQUENTIAL OMISSION WITHIN ALLOTTED TIME

------STEP:  3
  THIS STEP IS DIRECTLY LINKED TO STEP  1
          SUBLINK:A
            CAUSE:SC  --UNAWARE OF REQUIREMENT
   PRIMARY SYSTEM:PO  --OPERATION ACTIVITY
        COMPONENT:PLO --LICENSED OPERATOR PERSONNEL
   DIFFERENTIATOR:B   --
           EFFECT:TC  --CONSEQUENTIAL OMISSION WITHIN ALLOTTED TIME

------STEP:  4
  THIS STEP IS DIRECTLY LINKED TO STEP  1
          SUBLINK:A
            CAUSE:SC  --UNAWARE OF REQUIREMENT
   PRIMARY SYSTEM:PA  --ADMINISTRATIVE ACTIVITY
        COMPONENT:PUX --OTHER/ UNKNOWN UTILITY PERSONNEL
   DIFFERENTIATOR:C   --
           EFFECT:TB  --CONSEQUENTIAL OMISSION OF TASK, ANALYSIS, OR STE

------STEP:  5
  THIS STEP IS DIRECTLY LINKED TO STEP  1 AND STEP  2
                                AND STEP  3 AND STEP  4
            CAUSE:RS  --RESULTANT SUBSYSTEM FAULT
   PRIMARY SYSTEM:IU  --REACTOR PROTECTION
 SECONDARY SYSTEM:AD  --REACTOR VESSEL
        COMPONENT:1XZ --ONE OUT OF AN UNKNOWN NUMBER OF TRAINS
            TRAIN:Z   --UNKNOWN
           EFFECT:YB  --INFORMATION, CONSEQUENTIAL TO SEQUENCE

INITIAL UNIT CONDITIONS: E    STEADY-STATE OPERATION
            UNIT EFFECT: XX   NO SIGNIFICANT EFFECT

  EFFECT ON ENVIRONMENT: N    NO RELEASE
    EFFECT ON PERSONNEL: N    NO EXPOSURE
```

This series of commands (FIND, LOCATE, and XFIND) created three group numbers: group 1 with 6 LERs, group 2 with 2 LERs, and group 3 with 40 LERs. As you can see, the XFIND command (unlike the FIND command) does not reinitialize the group number to 1. Instead, the XFIND command will search the entire SCSS database, retrieve a specific group of LERs, and then give that set of LERs the next sequential group number (in this example, 3).

Having created these collections of LERs, we will now combine them, using both the COMBINE and MATCH commands, to create additional groups of LERs. As you will see, the major difference between these two commands is that the group of LERs output from the COMBINE command will contain all the LERs from either of the two groups (eliminating duplicates in the process), whereas the group of LERs output from the MATCH command will contain only those LERs that were contained in both original groups. Starting with the groups created with the FIND, LOCATE, and XFIND commands above, the following simulated screen displays the sequence of commands used to COMBINE and MATCH them.

```
*
COMBINE 1 3

==>GROUP 4 HAS      44 LERs--OPERATION: "COMBINE"
*
MATCH 1 3

==>GROUP 5 HAS       2 LERs--OPERATION: "MATCH"
```

As the first sequence shows, group 1 (with 6 LERs) and group 3 (with 40 LERs) have been joined with the COMBINE command to form group 4 (with 44 LERs); 2 duplicate LERs were eliminated. When groups 1 and 3 were joined with the MATCH command, only 2 LERs were output to group 5; only 2 LERs were contained in both group 1 and group 3. As you can see, the MATCH command is helpful in locating LERs when you have used different searching strategies and wish to determine areas of overlap.

At this time, you have five different groups of LERs that are potentially available for additional analysis. If you do not want to work any further with the active record group (group 5 in this example) you can use the RECALL command to "bring up" another group of LERs. The RECALL command allows you

to retrieve any previous group that was created by a search. The command syntax is very simple, requiring only the command and the group number you wish to retrieve. In the following example, you will use the RECALL command to retrieve a previously created group and then display it.

```
*
RECALL 4

==>LER GROUP 6 CONTAINS THE    44 LERs FROM GROUP 4
*
DISPLAY

    1 325/81-092     2 325/81-093     3 293/82-027     4 395/82-049
    5 416/83-192     6 389/83-052     7 213/81-002     8 219/81-025
    9 296/81-045    10 320/81-007    11 328/81-094    12 333/81-053
   13 334/81-070    14 338/81-037    15 339/81-008    16 369/81-037
   17 219/82-038    18 219/82-063    19 254/82-011    20 281/82-039
   21 298/82-023    22 311/82-057    23 324/82-139    24 338/82-088
   25 346/82-012    26 369/82-051    27 373/82-046    28 387/82-008
   29 416/82-019    30 325/83-025    31 362/83-002    32 362/83-081
   33 389/83-001    34 395/83-104    35 416/83-065    36 328/83-151
   37 370/83-082    38 416/83-183    39 346/83-072    40 370/83-062
   41 255/84-001    42 370/84-002    43 312/84-016    44 348/84-012
```

Because the LERs displayed in this example are not sorted in any particular order, use the KEYSORT command to display the LERs in consecutive order for ease of review. This command will sort the records in the active record group by docket number, year within docket number, and finally by LER number within year. The simulated screen illustrates the commands to be entered. Notice that the KEYSORT command does not display the LERs; the DISPLAY command must also be entered.

```
KEYSORT
*
DISPLAY
    1 213/81-002     2 219/81-025     3 219/82-038     4 219/82-063
    5 254/82-011     6 255/84-001     7 281/82-039     8 293/82-027
    9 296/81-045    10 298/82-023    11 311/82-057    12 312/84-016
   13 320/81-007    14 324/82-139    15 325/81-092    16 325/81-093
   17 325/83-025    18 328/81-094    19 328/83-151    20 333/81-053
   21 334/81-070    22 338/81-037    23 338/82-088    24 339/81-008
   25 346/82-012    26 346/83-072    27 348/84-012    28 362/83-002
   29 362/83-081    30 369/81-037    31 369/82-051    32 370/83-062
   33 370/83-082    34 370/84-002    35 373/82-046    36 387/82-008
   37 389/83-001    38 389/83-052    39 395/82-049    40 395/83-104
   41 416/82-019    42 416/83-065    43 416/83-183    44 416/83-192
```

The last part of this tutorial will examine three other SCSS commands (VALUES, RELATE, and TREND) that are available for analyzing the active group of LERs. All three commands are used to examine and analyze the step matrix contained in the retrieved LERs. Enter the following sequence using the FIND, STEPSCAN, and VALUES commands.

```
FIND
+
<COMP> PLO PUX
+
END
 ==GROUP  1 HAS    78 LERs--OPERATION: "FIND"
*
STEPSCAN
+
<COMP> (PLO PUX) <EFF> TB
+
END
THE FOLLOWING "STEP" DATA COLLECTIONS WILL BE CAPTURED:
    THE STEPS SATISFYING THE CURRENT SEARCHING LOGIC
 ==GROUP  2 HAS    40 LERs OUT OF    78 SEARCHED--OPERATION: "
*
VALUES * <PSYS>
A "VALUES" ANALYSIS WILL BE PERFORMED FOR THE PSYS     FIELD
      40 OUT OF     40 LERs CONTRIBUTED TO THE ANALYSIS
THE ACTIVE LIST OF    40 LERs HAS     4 UNIQUE VALUES IN THE PSYS     FIELD
FOR STEPS THAT SATISFY THE CURRENT LOGIC TABLE
KEY VALUE  NUMBER OF STEPS                DESCRIPTION
---------  ---------------
  PO           24( 43.6%)   OPERATION ACTIVITY
  PT           15( 27.3%)   TEST/CALIBRATION ACTIVITY
  PA           15( 27.3%)   ADMINISTRATIVE ACTIVITY
  PM            1(  1.8%)   MAINTENANCE/REPAIR ACTIVITY
```

As you can see, the initial FIND command retrieved 78 LERs that contained both PLO and PUX values in the component field. The STEPSCAN command then narrowed the active group of LERs to those 40 that contained either the component code PLO in the same step with the effect code TB **or** the component code PUX in the same step with the effect code TB. The VALUES command was then used to examine the step matrixes in that group of 40 LERs retrieved with the FIND and STEPSCAN commands. In this example, we were interested in determining which primary systems (PSYS) were involved in this particular group of steps. As the table shows, 24 (43.6%) steps were involved with an OPERATION activity, 15 (27.3% steps were involved with both TEST/CALIBRATION activities and ADMINISTRATIVE activities, and only 1 1.8%) step was involved

with a MAINTENANCE/REPAIR activity. The VALUES command can also be used for other field designations. For example, enter the following command.

```
*
VALUES * <COMP>
A "VALUES" ANALYSIS WILL BE PERFORMED FOR THE COMP     FIELD
       40 OUT OF     40 LERs CONTRIBUTED TO THE ANALYSIS
THE ACTIVE LIST OF    40 LERs HAS     2 UNIQUE VALUES IN THE COMP     FIELD
FOR STEPS THAT SATISFY THE CURRENT LOGIC TABLE
KEY VALUE  NUMBER OF STEPS                DESCRIPTION
---------  ---------------
  PLO          31( 56.4%)   LICENSED OPERATOR PERSONNEL
  PUX          24( 43.6%)   OTHER/ UNKNOWN UTILITY PERSONNEL
```

As this example shows, the licensed operators were involved in 31 (56.4%) steps and other unknown personnel were involved in 24 (43.6%) steps in this group of 40 LERs. This VALUES analysis does not, however, tell us which primary systems were involved with each type of personnel. To do this type of analysis, we need to use the RELATE command. Comparing the output from the two VALUES commands to that from the RELATE command will demonstrate the different uses of the two commands.

The RELATE command is used to find all unique pair combinations of key codes for any two fields in the coded step matrix. The output is a two-dimensional table that gives the counts in each cell. The manner in which this command is entered is similar to the VALUES command, except two fields must be specified for analysis. Searching logic must also be included in RELATE; however, we will use the same logic that was used with the VALUES command, telling the computer that we want to use the same logic by entering an asterisk (*) in the command line following the command name. The following simulated screen displays the commands and the output for creating a table relating the primary system field with the component field.

```
RELATE * <PSYS> <COMP>

"RELATE" WILL ANALYZE COMBINATIONS OF CODES IN THE PSYS     AND COMP     FIELDS
FIELD:PSYS     HAS   4 UNIQUE VALUES
FIELD:COMP     HAS   2 UNIQUE VALUES

*PSYS
 *
  *
   *
    *
     *
      *
       *

COMP     *  PO     PT     PA     PM
---------+------+------+------+------+-------+
   PLO        20      7      4      0     31
---------+------+------+------+------+-------+
   PUX         4      8     11      1     24
---------+------+------+------+------+-------+
SUM           24     15     15      1     55
```

As you can see, the licensed operators (PLO) were involved in 20 steps dealing with operation activities (PO), whereas unknown personnel (PUX) were mostly involved with maintenance and repair activities (PA). The RELATE command, like the VALUES command, can be used with different combinations of fields in the step matrix.

TREND is an SCSS command that produces a two-dimensional table showing how all unique key codes for a particular field correspond with the event dates of the LERs. This command is similar to the VALUES and RELATE commands, and the logic specified for the previous analyses will be used (specified with the asterisk). In this example, the TREND command will request an analysis of primary system (PSYS) values.

```
TREND * <PSYS>
A "TREND" ANALYSIS WILL BE MADE FOR THE PSYS     FIELD
THE TIME INTERVALS WILL BE BY QUARTER STARTING WITH 01/01/81
FIELD:PSYS     HAS   4 UNIQUE VALUES

*PSYS
 *
  *
   *
    *
     *
      *
       *
```

B DATE *	PO	PT	PA	PM	
01/01/81	3	2	0	0	5
04/01/81	2	0	1	0	3
07/01/81	2	1	2	0	5
10/01/81	0	0	1	0	1
01/01/82	2	1	0	0	3
04/01/82	3	2	1	0	6
07/01/82	1	1	1	0	3
10/01/82	2	1	3	0	6
01/01/83	0	0	1	0	1
04/01/83	2	1	1	0	4
07/01/83	2	0	0	1	3
10/01/83	4	3	3	0	10
01/01/84	1	2	0	0	3
04/01/84	0	1	1	0	2
SUM	24	15	15	1	55

This table shows that the primary system value PO is involved in most of the steps and that the occurrence of the different systems involved has been fairly constant over time. A similar type of TREND analysis could also be conducted for the component field or any other field in the step matrix.

TERMINATING THE SESSION

The last command introduced is the STOP command, which is used to terminate the SCSS session. It can be used anytime throughout the session when you are at the command level (when you are prompted by an asterisk). The following simulated screen illustrates this command and the computer response.

```
*
STOP

THIS SESSION HAS USED 8.03 SECONDS OF CPU TIME AND
HAS BEEN ACTIVE FOR 2903.43 SECONDS
```

This concludes the tutorial section of the User's Guide. This chapter is intended to illustrate the various uses of the SCSS and give you an indication of the comprehensiveness and complexity of the types of analyses that can be conducted on this database. Detailed descriptions and further examples of the commands used in the tutorial, along with more advanced commands, are presented in Chaps. 3 and 4.

3. FUNDAMENTAL AND HELP COMMANDS

3. FUNDAMENTAL AND HELP COMMANDS

3. FUNDAMENTAL AND HELP COMMANDS

This chapter outlines the function and syntax of the commands that are fundamental in using the SCSS. Figure 3-1, SCSS Command/Field Cross Reference Chart, lists the SCSS fields and indicates the commands that can use a given field. This chapter discusses the HELP commands first so that you will have them available as you begin your searching. All other commands are discussed in the order that you are likely to use them; for instance, FIND is usually the first command issued during a session, so it is discussed first. Logging onto and exiting the system are detailed in Chap. 2, as are the basic steps to be performed when using the commands. This chapter gives you more information about each of the commands and discusses options and alternatives in detail.

Options 2, 3, and 4 given under the FIND command are particularly significant. These options, described in detail for FIND, are also applicable to other commands that define searching strategy.

- Option 2 explains how to use the "and/or" logic of SCSS.
- Option 3 introduces relational symbols that can be used with several other commands as well.
- Option 4, the FIND HELP command, is invaluable in allowing you to use SCSS without having to refer to Sequence Coding and Search System Coder's Manual for Licensee Event Reports - Code Listings, ORNL/NSIC-189.

A table listing all SCSS commands and allowable abbreviations (if applicable) is included as a reference at the end of this chapter.

NOTE: In Chaps. 3 and 4, words that you should enter are shown in **boldface type** just as they are in the tutorial. However, so that you can quickly distinguish a command from a variable, commands are shown in uppercase letters (e.g., **FIND**) and variables are shown in lowercase letters (e.g., for **<field>value** you might type **<comp>dsl**).

FIGURE 3-1. SCSS COMMAND/FIELD CROSS-REFERENCE CHART

FIELD	DESCRIPTION	Commands that can use the field			
		FIND XFIND	SCAN STEPSCAN LINK JOIN	VALUES RELATE TREND	SVALUES SRELATE STEPSORT
DOCKET	Docket Number			X	X
YEAR	Year			X	X
TYPE	Reactor Type			X	
REGION	Region			X	
NSSS	NSSS Supplier			X	
AE	Architectural Engineer			X	
SYMBOL	Facility Operator Symbol			X	
WATCH	Watch-List Code	X			
STEP	Step Number				X
LK	Link		X	X	X
SLK	Sublink		X	X	X
CAUSE	Cause	X	X	X	X
PSYS	Primary System	X	X	X	X
ISYS	Interface System	X	X	X	X
COMP	Component	X	X	X	X
VEND	Vendor	X	X	X	X
QUAN	Quantity		X	X	X
TR	Train		X	X	X
CH	Channel		X	X	X
DI	Differ		X	X	X
T	Timing	X	X	X	X
P	Performance	X	X	X	X
D	Detection	X	X	X	X
EFF	Effect	X	X	X	X
ICOMP	Coarse Component Code	X	X	X	X
ISEQ1	Sequence Number in Event				X
ISEQ2	Step Type				X
ISEQ3	Sequence Type				X

HELP

Function

You can request general assistance when you are in SCSS by typing HELP in response to an asterisk (*) system prompt. After you type HELP and depress the ENTER key, your screen will display the SCSS Command Chart, shown in Fig. 3-2. If you then type HELP and a particular command, the screen will next display a summary of that command's functions, allowable abbreviations, and a brief description of the syntax.

Syntax

HELP command

Example

```
*
HELP FIND
```

The FIND command is normally the first command that will be executed in a sequence of commands to locate LERs. The FIND only works with inverted fields. The LERs found by this command will be identified as "active record group" 1. In order to locate LERs without initializing the group counter, the XFIND command is provided.

Allowed Abbreviations: FI or F

Syntax: FIND
(In this case, the search logic is specified on subsequent lines as described in the SYNTAX help file)

or: FIND *
(In this case, the FIND uses the logic which was constructed for the last searching operation)

or: FIND ALL
(This form locates all records in the database)

or: FIND HELP
(This form is used to interact with the authority files to construct searching logic)

Fields which can be used with a FIND are:

CAUSE, PSYS, ISYS, COMP, VEND, T, P, D, EFF, ICOMP, and WATCH.

Note that a FIND operates on the LER in a global sense, wherein each field is treated as a multiple entry field. If one asks for LERs with a particular system (PSYS) key code and a particular component (COMP), the resultant LERs will contain both key codes, but not necessarily on the same step. A FIND should never be used in the "not equal" sense since the multiple entry characteristics of the LER fields will almost certainly always satisfy this kind of screening.

FIGURE 3-2. SCSS COMMAND CHART

1. Locate the Initial Group of LERs in a Sequence of Searching Operations.

   FIND, XFIND, BUILD, ACTIVE

2. Screen a Group of LERs.

   SCAN, STEPSCAN, LINK, JOIN, TEXTSCAN, LOCATE

3. Process a Group of LERs.

   VALUES, RELATE, TREND, REF

4. Process "Captured" Step Data Collections.

   STEPLIST, SVALUES, SRELATE, STEPSORT

5. List Record Keys and LERs.

   LIST, OPTIONS, KEYSORT, DISPLAY

6. Manipulate and Combine Record Groups.

   COMBINE, MATCH, NOMATCH, NOTFOUND, RECALL, REMOVE

7. Display or Edit Searching Logic.

   TABLE

8. Request Assistance.

   HELP, DOCKET, INVERTED, AUTH

9. Change the Record Searching Hierarchy.

   DEFINE, DEFAULT

10. Manipulate Saved Searching Strategies.

    ACTIVE, SAVE, DELETE, EXEC, DIRECTORY

11. Terminate this Session.

    STOP

Enter HELP HELP for more information on the commands or enter

HELP Command

for detailed information on an individual command.

Additional aid can be obtained by typing:

HELP SYNTAX -- describes syntax for searching commands
HELP FIELDS -- lists available fields
HELP FILES -- lists files in the SCSS System

AUTH

Function

An SCSS authority file contains decoded text for the codes used in the database. The AUTH command allows searching the decoded text strings in order to locate codes. This useful option is analogous to the operations performed in the FIND HELP command discussed in the tutorial, except AUTH does not search the database for LERs, but rather screens decoded text on a word-by-word basis in an attempt to locate an applicable code. In a sense, this command is like using an inverted dictionary, wherein you specify the "meaning" to retrieve the "word."

Syntax

AUTH <field> word1 word2 word3 etc.

This form outputs codes containing all the words (in any order) in the authority records for the particular field. If you specify a field but no words, the computer will produce a table with all codes for the field you specified. However, use this form with discretion because some of the fields contain several hundred codes.

AUTH word1 word2 word3 etc.

If you do not specify a field name, AUTH will scan the entire authority file (except for the vendor codes; see note below) and give you all possible codes for the words you have specified.

AUTH <field> OR or **ORDER word1 word2 word3 etc.**

By default, all of the words on the AUTH command line must exist in the authority file record to produce a hit, but these words can occur in any order. If you include the term **OR** on the command line, all codes that contain any of the words will be listed. If you include the term **ORDER** on the command line, all the words must exist in the order in which they are specified. The field name must immediately follow the word AUTH, and OR or ORDER must immediately follow the field name (i.e., as the third word in the command), or they will be treated as words to use in the text string searching.

NOTE: The VEND (vendor) field is treated in a different manner from other SCSS fields. By default, only vendor codes that begin with the initial letter of the first word of text in the command will be screened. However, you can include the term **LETTER=ALL** in the AUTH command line to tell the computer to search vendor codes beginning with all letters, or you can include **LETTER=x** to have the computer search codes beginning with the letter you specify.

Example 1 *

AUTH <comp> personnel

This example asked for all codes that contain the word personnel in the COMP field authority records. Once you know the code you want, you can use the FIND command to locate the LERs that may be of interest to you.

Example output is shown on p. 2-16.

Example 2 *

AUTH <vend> Westinghouse

```
FIELD: VEND     VAR: W119 -- WESTINGHOUSE AIR BRAKE COMPANY (WABCO)
FIELD: VEND     VAR: W120 -- WESTINGHOUSE ELECTRIC CORP.
FIELD: VEND     VAR: W121 -- WESTINGHOUSE ELECTRIC COMPANY (ELEV.DIV)
FIELD: VEND     VAR: W123 -- WESTINGHOUSE ELECTRIC SUPPLY COMPANY
FIELD: VEND     VAR: W351 -- WESTINGHOUSE ELEC CORP.-NUCLEAR ENERGY SYS
```

The program considered only those codes in the vendor field that begin with the letter "w".

Example 3 *

AUTH <vend> estinghouse LETTER=w

Although in this case, this command would produce the same output as shown in example 2, the computer found the output in a different manner. Instead of searching for vendor codes beginning with "w", it searched for the text string "estinghouse" in vendor codes beginning with "w".

DOCKET

Function

SCSS includes a special authority file that contains data related to the various reactor facilities (or dockets). The DOCKET help command allows searching this file to locate facilities that satisfy some criteria. The DOCKET help command can be used in one of two ways. First, you can enter a docket number, and the computer will display information associated with that plant. Second, you can enter one or more words to locate docket records that contain those words in their textual fields. You can search the file based on information other than the docket number, such as reactor type, NRC region, NSSS supplier, architectural engineer, and symbol (code) for the facility operator. (Note, though, that in the second form the command treats the two text fields for facility name and operator as one field.)

Like the AUTH command, DOCKET searches code tables instead of searching the database for LERs.

Syntax

DOCKET nnn

This form locates the record for the docket numbered nnn.

DOCKET word1 word2 word3 etc.

This form locates the docket records that contain all words in the record's textual information.

DOCKET <field>=value

This form locates docket records that have the value in the field. To search on

reactor type, include **TYPE=ttt**, where ttt is PWR or BWR
region, include **REGION=r**, where r is one of the five NRC geographical regions
NSSS supplier, include **NSSS=vvvv**, where vvvv is the code for the NSSS supplier
architectural engineer, include **AE=aaaa**, where aaaa is the appropriate code
symbol of the facility, include **SYMBOL=sss**, where sss is the appropriate code

Example 1 •

```
DOCKET TYPE=bwr REGION=5

DOCKET:133  HUMBOLDT BAY                          TYPE:BWR
                  REGION:  5                      NSSS:GE
ARCHITECTURAL ENGINEER:  BECH
     FACILITY OPERATOR:  PACIFIC GAS & ELECTRIC CO.
                  SYMBOL:  PGE
```

Shown is one of the three entries located when this command was issued.

Example 2 •

```
DOCKET three mile island
```

The computer will search for the text string Three Mile Island. Example output is shown on p. 2-16.

INVERTED

Function

The INVERTED command accesses the 11 inverted fields (explained in Chap. 1), produces counts of the use of codes in the entire database, and decodes the codes in its output. INVERTED has an option that allows you to list only the codes that have the most uses in the database for a particular field. By default, the lines in the table are sorted by count in descending order; however, including the term **KEYSORT** anywhere in the command line causes the lines to be sorted alphabetically by code. INVERTED can also be used to report on the number of uses of a particular code by including the code after the field name in the command line.

Syntax

INVERTED <field>

This lists the counts for codes used in the field you specify.

INVERTED <field>=value

This lists the counts for a particular code in a field.

INVERTED TOP n <field>

If you include **TOP n**, where n is a number, the computer will list only the codes that have the n highest counts in the field.

INVERTED <field> KEYSORT

Instead of being sorted by count, the output will be sorted alphabetically by code.

Example

```
*
INVERTED <cause> TOP 10

FIELD VAR  LERS  STEPS              DESCRIPTION
----- ---- ----- -----
CAUSE RC   10661 22698  RESULTANT COMPONENT FAULT
CAUSE ZZ    5842  7504  UNKNOWN
CAUSE RS    3413  4573  RESULTANT SUBSYSTEM FAULT
CAUSE RT    1942  2613  RESULTANT TOTAL SYSTEM FAULT
CAUSE SZ    2368  2521  UNKNOWN PROCEDURAL/GENERAL HUMAN FACTORS
CAUSE LA    1095  1202  DRIFT/OUT OF CALIBRATION
CAUSE SH     929   996  ERROR IN REQUIREMENT
CAUSE SX     648   701  OTHER PROCEDURAL/GENERAL HUMAN FACTOR
CAUSE PD     616   694  CRUD BUILDUP
CAUSE XX     547   654  OTHER
```

This command produced a table showing the number of times the 10 codes used most often in the cause field occurred. The table is sorted alphabetically by code.

FIND

Function

The FIND command locates (but does not read) LERs based on screenings of codes in the 11 inverted fields (discussed in Chap. 1). These fields are:

CAUSE (Cause)
PSYS (Primary System)
ISYS (Interface System)
COMP (Component)
VEND (Component Vendor)
T (Timing)
P (Performance)
D (Detection)
EFF (Effect)
ICOMP (Coarse Component Code)
WATCH (Watch-List Code)

The FIND command locates LERs from the entire database that satisfy the searching strategy that you specify. These LERs become "active record group 1."

Subsequent processing searches records from this record group in a top-down fashion; each succeeding record group will be a subset of its predecessor. The exception to this is when the XFIND command (discussed next) is used. In XFIND, the record group is the next available record group, and these LERs become the collection available for processing and further screening.

The FIND command operates at the LER level, which means that you can ask for more than one value in a field and for more than one field.

Syntax

FIND

When you type FIND in response to the asterisk prompt, the computer will display the plus prompt, asking you to supply the information for the FIND search. Enter the appropriate searching syntax lines (then press the ENTER, or RETURN, key). Example alternatives are given below. When you have finished giving all your instructions, type **END** to tell the computer to execute the search. If you decide that you do not wish to execute the search and want to enter a new command, type **ABORT** to tell the computer to ignore the search.

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 32 LERs--OPERATION:  "FIND"
```

This example asked the computer to find LERs referencing diesel engines as components. The computer found 32 LERs

that satisfy this criterion. You can use the LIST command to have these LERs displayed, or you can issue another command to narrow your active record group.

Alternative 1: Multiple Values in a Field

You can ask for multiple values in a field by giving the values one after the other with intervening blanks.

<field> value value value

If you specify

<comp> dsl mot zvz

the computer will find LERs containing component codes for diesel engines (dsl), motors (mot), and valves of unknown size and material (zvz) all in the same LER.

Alternative 2: Multiple Fields

You can also specify multiple fields on the same line.

<field> value <field> value

If you enter

<comp> dsl <cause> ha

the computer will find LERs containing a component code for diesel engines and a cause code for high temperature (ha). (Note that in the case of the FIND command, the component code dsl and the cause code ha do not have to be on the same step.)

OPTIONS FOR THE FIND COMMAND

Option 1: FIND ALL

Function This one-line command locates all records in the database. You will usually need a more specific command.

Syntax

```
FIND ALL
```

Option 2: Boolean Logic (And/Or Searching)

Function The FIND, SCAN, STEPSCAN, LINK, JOIN, VALUES, RELATE, and TREND commands use the syntax described below.

The logic is grouped into one or more packets called "cases." For most commands the logic between cases is "or"; i.e., satisfying case 1 or case 2 or case 3, etc., produces a hit.

A case is made up of one or more "or-groups." The logic between or-groups is "and"; i.e., satisfying a single case requires that its first or-group and its second and its third, etc., be satisfied.

An or-group is made up of one or more "screens." A screen is the smallest element in the logic and is used to make one kind of comparison about a value in a particular field; for example, asking that the component field contain the code for motors is one screen. In most instances, an or-group will consist of one screen. The logic between screens (if the or-group contains more than one screen) is "or"; i.e., satisfying an or-group requires satisfying only one of its screens.

Logic Hierarchy:

Case 1 or Case 2 or Case 3 or ...
Or-Group 1 and Or-Group 2 and Or-Group 3 and ...
Screen 1 or Screen 2 or Screen 3 or ...

NOTE 1: The syntax described in options 2, 3, and 4 can be used with many of the LER searching and processing commands in SCSS, including FIND, SCAN, STEPSCAN, LINK, JOIN, VALUES, RELATE, and TREND.

NOTE 2: If you need more than one line (72 characters) to specify a case, you can continue the case on subsequent lines by putting a plus (+) as the first character on each subsequent line. (This should not be confused with the + used as a prompt for more input.) However, do not split a screen across lines.

NOTE 3: SCSS allows you to use only one level of parentheses, but you can use several nests within a case, if necessary.

NOTE 4: The terms "and" and "or" never appear on the command line; however, the relational symbols discussed in option 3 (p. 3-18) allow you to establish relational connections.

Syntax—Use of Multiple Lines

Within each case (between or-groups), the logical connection is "and," unless you use parentheses (discussed next). When you want the computer to perform a search using a logical operator of "or," you can tell it to do so by entering separate cases on separate lines, as shown below.

```
+
<field> value
+
<field> value
+
<field> value
+
END
```

Example

```
+
<comp> dsl
+
<comp> mot
+
<comp> zvz
+
END
==>Group 1 HAS 1118 LERs--OPERATION:  "FIND"
```

This example asked the computer to find LERs with component codes involving diesel engines (dsl) or motors (mot) or valves of unknown type and material (zvz). Three cases are involved, each consisting of one or-group containing one screen.

Syntax—Use of Parentheses

If you need "or" logic within a case, you must use parentheses to set off those field-value pairs that need "or" connections. If you use parentheses, a single case can accomplish what the three cases with "or" connections did in the previous example.

```
<field> (value1 value2 value3)
```

Example 1

```
*
FIND
+
<comp> (dsl mot zvz)
+
END
==>GROUP 1 HAS 1118 LERs--OPERATION "FIND"
```

This example also asks for LERs involving a component code of dsl or mot or zvz. The single case has one or-group consisting of three screens.

Example 2 You can specify multiple fields within the "or" range (inside the parentheses).

```
*
FIND
+
<comp> (dsl <cause> xx yy)
+
END
==>GROUP 1 HAS 1051 LERs--OPERATION "FIND"
```

This example asks for a component code of dsl or a cause code of xx or a cause code of yy. The single case has one or-group that contains three screens.

Example 3 You can use variations of this syntax depending on your own needs; however, it is important to remember that although you can use only one level of parentheses, you can use as many or-groups as you need to define a case. Therefore, if you want the computer to find a component code of diesel engines or motors or a cause code of xx or yy, you can type

```
<comp> (dsl mot <cause> xx yy)
```

Example 4 Because you can use only one level of parentheses, you may need multiple lines to identify the exact search you want. To find LERs involving a component code of dsl or LERs including a cause code of xx and yy, you can type

```
+
<comp> dsl
+
<cause> xx yy
+
END
```

A line reading

```
<comp> (dsl <cause> xx) yy
```

however, would locate LERs with a component code of DSL and a cause code of yy or LERs with a cause code of xx and a cause code of yy.

Syntax—Referencing by Position

When you need to address a particular location within a field's code, you can reference the position by including it in enclosed parentheses after the field name.

Example The pipe component codes all contain PI as the first two characters. The third character is used to designate a size class, and a fourth character designates the material from which the pipe is made. To find LERs involving pipes made from carbon steel (fourth character is a C) you can type

```
<comp> .eq. pi <comp(4)> .eq. c
```

This single case has two or-groups consisting of one screen each. (See also Option 3: Relational Searching.)

Option 3: Relational Searching

Syntax

<field> .relational symbol. value

Relational searching allows you to search for a relationship in addition to an exact match. Ten relational symbols are available to you.

Symbol	Relationship
.EQ.	is equal to
.GE.	is greater than or equal to
.GT.	is greater than
.LE.	is less than or equal to
.LT.	is less than
.NE.	not equal
.BEG.	begins with
.CT.	contains
.NBEG.	does not begin with
.NCT.	does not contain

NOTE: Any relational symbol must be preceded and followed both by a space and a period (.).

Example

```
*
FIND
+
<comp> .beg. z
+
END
==>GROUP 1 HAS 916 LERs--OPERATION "FIND"
```

This example specifies a case that requests component codes beginning with a "Z" (which denotes an unknown component type).

Option 4: Interactive Searching (FIND HELP)

Syntax

FIND HELP

At the + prompt, enter

<field> word1 word2 word3 etc.

The computer will supply you with a list of choices. Enter **SELECT** and the code(s) you want the FIND to search for, or enter the next field selection line if you want all of the codes found to be used in the search. If it is more convenient, you can enter **DELETE** and the code(s) you do not want to search for, or you can enter **DELETE ALL** to ignore all codes.

This option allows you to enter text strings instead of codes. When you type in the word (or words) you would like to find, the computer will display the symbols for all definitions that contain that word. You will then determine the codes you want. This option, basically, combines the AUTH command with the normal FIND command.

Example 1

```
*
FIND HELP
+
<comp> diesel
SYMBOL:DSL  ENGINE, DIESEL
+
<comp> motor
SYMBOL:MCC  CONTROL CENTER, MOTOR
SYMBOL:MG   GENERATOR, MOTOR
SYMBOL:MOT  MOTOR
SYMBOL:MSTR MOTOR STARTER
+
SELECT mot
+
END
==>GROUP 1 HAS 46 LERs--OPERATION:  "FIND"
```

Because only one code exists for "diesel," the computer did not require a selection. When the computer finds two or more codes, you can select one or more of the terms (as in the example), or you can delete one or more of the terms (using DELETE and naming the terms) or all of the terms (using DELETE ALL to delete all entries).

NOTE: Because of the unique features of the FIND HELP option, the logical connection between lines is "and" until you specify that a new case is beginning. Note that there are no provisions at present for explicitly specifying SCSS codes, as you can do with other FIND options. In the preceding example, the computer would search for LERs with component codes of diesel and motor. However, the logical connection is automatically "or" when the computer finds more than one code. If a motor code had not been

specified in the example, the logic would have been to find LERs involving diesels and one or more of the following codes: motor control centers (mcc) or motor generators (mg) or motors (mot) or motor starters (mstr).

To initiate a new case enter

CASE

then describe the logic for the case. The logical connection between cases is "or."

Example 2

```
*
FIND HELP
+
<comp> diesel
SYMBOL:DSL    ENGINE, DIESEL
+
CASE <comp> motor
SYMBOL:MCC    CONTROL CENTER, MOTOR
SYMBOL:MG     GENERATOR, MOTOR
SYMBOL:MOT    MOTOR
SYMBOL:MSTR   MOTOR STARTER
+
END
==>GROUP  1 HAS  629 LERs--OPERATION:  "FIND"
```

In example 1, a similar series of commands located only 46 LERs. Entering CASE, however, established a logical connection of "or" (instead of "and"), and 629 LERs were located. Note also that the computer searched for all the codes for motor because no selection was made.

Example 3

```
*
FIND HELP
+
<comp> sensing line
SYMBOL:SL  SENSING LINE
+
<cause> frozen
NO MATCHES ARE IN THE AUTHORITY FILES FOR YOUR QUERY

+
<cause> freeze
NO MATCHES ARE IN THE AUTHORITY FILES FOR YOUR QUERY

+
<cause> freez
SYMBOL:HC  FREEZING CONDITION
+
END
==>GROUP 1 HAS 20 LERs--OPERATION "FIND"
```

In this example, the computer did not find the cause code for freezing condition immediately; consequently, it asked for alternative text strings until it found a match. If the category is not provided in the SCSS codes or if the text in the authority file is incomplete for the code (which you can assume if you try every iteration you can think of without achieving a match), you must type ABORT instead of END to be returned to the command entry level.

XFIND

Function The XFIND command locates LERs based on screenings of the codes in the 11 inverted fields (see Chap. 1). These fields are:

CAUSE (Cause)
PSYS (Primary System)
ISYS (Interface System)
COMP (Component)
VEND (Component Vendor)
T (Timing)
P (Performance)
D (Detection)
EFF (Effect)
ICOMP (Coarse Component Code)
WATCH (Watch-List Code)

XFIND is exactly like the FIND command except for one subtle departure. Whereas FIND always initializes the searching strategies and locates active record group 1, XFIND locates a group that becomes the next available group number in the sequence. Subsequent processing is on this new active group, in the same top-down fashion as with the FIND. Groups formed earlier from a FIND or other XFINDs remain available for processing (for example, for combining with groups from the new sequence).

All syntax and options for XFIND are identical with those of the FIND command discussed previously.

DISPLAY

Function The DISPLAY command lists record keys (i.e., docket number, year, and LER number) from the active record group. DISPLAY gives you sets of numbers (sequential identifiers) that can be used to call for the LER record you want to view (see LIST command). If a record group contains so many keys that listing them would produce lengthy output, the computer will prompt you to determine whether the request should be executed or ignored.

Syntax

DISPLAY

DISPLAY filename

In this case, the keys for the record group located by the searching strategy named "filename" (saved along with a previously stored group of LERs using the SAVE command) are displayed.

Example

```
* DISPLAY

1 416/82-033    2 416/82-074    3 416/82-097    4 373/83-072
5 416/83-049    6 373/83-132
```

In the first item in this example, 1 is the sequential identifier within the active group, 416 is the docket code, 82 is the year (1982), and 033 is the LER number.

LIST

Function The LIST command is used to display an LER record. You can select which portions of the LER record to display by using the OPTIONS command (discussed next). You can select the LER using either the computer-assigned identifier of an active group (1 through 6 in the example given with the DISPLAY command) or the docket number, year, and LER number (substituting spaces for the slash and hyphen). The LIST command simply displays LERs.

Syntax

LIST n

This syntax lists the LER record using the computer-assigned identifier (n) of the active record group.

LIST ddd yy nnn

This syntax lists the LER record using the docket number (ddd), the year (yy), and the LER number (nnn).

LIST n1 n2

This syntax allows you to list more than one LER record using a single command. n1 and n2 are the first and last identifiers of the inclusive group that you want listed from the active record group. (If you request listing more than 10 LERs at a time, the computer will question you to ensure the request was intentional to avoid consuming considerable time and paper.)

Example

```
*
LIST 1
```

or

```
*
LIST 325 81 093
```

Either of these commands tells the computer to list an LER; in the first case, it is the first LER in the active record group. In the second case, it is a specific LER. The output will depend on the options selected when the command is issued.

Example output is shown on p. 2-24.

OPTIONS LIST

Function The default options for the LIST command display the entire LER, with the exception of the coded step matrix. The information in each LER is divided into eight categories: (1) header information, (2) comments, (3) docket information, (4) watch-list codes, (5) reference LERs, (6) the coded step matrix, (7) the abstract, and (8) reportability requirements. If you do not need to see the entire LER, you can select only the portions you do need to see by using the OPTIONS LIST command. Please note: the header information cannot be suppressed; it is always displayed to identify the LER. When you use the OPTIONS LIST command to prevent certain categories of information from being displayed on your screen, the categories you specified will remain suppressed until (1) you select an option's counterpart or (2) you enter OPTIONS LIST, with no options specified, which will return all options to their defaults.

Syntax

OPTIONS LIST option1 option2 etc.

The available options are listed in the following table.

Category Affected or Action	Command to Suppress	Command to Display
Comments	nocomm	comments
Docket Information	nodock	docket
Watch-List Codes	nowatch	watch
Reference LERs	noref	ref
Coded Step Matrix	nomat	matrix
Abstract	noabst	abstract
Decode (do not decode) the step matrix	nodec	decode
Include (suppress) the timing, performance, and detection fields when decoding the step matrix	notpd	tpd
Produce (do not produce) a flow diagram for the step matrix	noflow	flow

The last three options in the table are special options that produce additional information concerning an LER. These options can decode the step matrix, include or suppress three fields in the decoded step matrix, and produce a flow diagram of the step matrix (as described in Chap. 4). The decode option is particularly helpful because it will translate the codes used in the step matrix. The output, however, is quite lengthy, so the decode option should be used with discretion.

Two more forms of the OPTIONS command are provided to assist you.

OPTIONS LIST

If you simply type this command without specifying any options, the computer will return you to the default set of options.

OPTIONS LIST ?

The screen will show a list of currently active options.

NOTE: Once set, an option remains active until cancelled or reset.

Example

```
*
OPTIONS LIST NOCOMM NOREF NOWATCH NOABST DECODE MATRIX
*
LIST 325 81 093
```

After this input, the screen will show the header information, docket information, coded step matrix, and the decoded step matrix for LER 325/81-093. It will not display any comments, reference LERs, watch-list codes, the abstract, or a flow chart.

Example output is shown on pp. 2-26 and 2-27.

LOCATE

Function The LOCATE command scans the LERs in the active record group and screens them against information contained in the docket file to select LERs that meet criteria contained in the docket file. For example, after you have found a group of LERs using the component code for diesel engine (dsl), you may need to locate only those LERs that apply to a particular docket, facility operator (such as TVA), year, architectural engineer, region, NSSS, or reactor type. You can repeat the LOCATE command if you need to further narrow your active record group. At present, nine fields can be used with the LOCATE command:

DOCKET	-	docket code (ddd)--multiple dockets allowed
NAME	-	name of the facility
OPERATOR	-	name of the facility operator
YEAR	-	year (yy)--multiple years allowed
TYPE	-	reactor type (PWR or BWR)
REGION	-	geographical region (r)
NSSS	-	code for the NSSS supplier (vvvv)
AE	-	code for the architectural engineer (aaaa)
SYMBOL	-	code for the reactor operator (sss)

NOTE: To use the LOCATE command, you must have an active record group. If you want to locate from the entire database, execute a FIND ALL command then execute LOCATE.

Syntax

```
LOCATE DOCKET=ddd YEAR=yy etc.
```

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 591 LERs--OPERATION   FIND"
*
LOCATE SYMBOL=tva TYPE=pwr
==>GROUP 2 HAS 25 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
```

This sequence located LERs involving diesels, but only at TVA's pressurized water reactors. Note that only the LERs in group 1 were examined; the entire database was not scanned.

LOCATE, CHART OPTION

Function The LOCATE CHART option allows a quick analysis of (1) the facilities that submitted the LERs contained in the active record group and (2) the years of the event dates. The CHART option produces a table that shows a count of LERs by facility name, docket number, and event year(s).

Syntax

LOCATE CHART

Example 1 This example uses group 2 from the LOCATE example on p. 3-27.

```
*
LOCATE CHART
==>GROUP 3 HAS 25 LERs OUT OF 25 SEARCHED--OPERATION: "LOCATE"

COUNT ON LERs SUBMITTED
```

FACILITY	DOCKET	1980	1981	1982	1983	1984	TOTAL
SEQUOYAH 1	327	0	6	3	11	0	20
SEQUOYAH 2	328	0	0	3	2	0	5
		0	6	6	13	0	25

Example 2

```
*
FIND ALL
==>GROUP 1 HAS 13652 LERs--OPERATION:  "FIND"

*
LOCATE CHART NAME=Millstone 1 YEAR=83
LERS FOR THE FOLLOWING YEARS WILL BE KEPT:  83
LERS WILL BE KEPT FOR THE FACILITY:  MILLSTONE 1
==>GROUP 2 HAS 35 LERs OUT OF 13652 SEARCHED--OPERATION: "LOCATE"
```

FACILITY	DOCKET	1980	1981	1982	1983	1984	TOTAL
MILLSTONE 1	245	0	0	0	35	0	35
		0	0	0	35	0	35

In example 2, the user needed to know the number of LERs submitted for a given plant, Millstone 1, in a given year, 1983.

SCAN

Function

The SCAN command allows you to search values in any field in the coded step matrix of an LER. Although SCAN is similar to the FIND command, SCAN can search in any field whereas FIND can search only in the 11 inverted fields. On the other hand, FIND is far more efficient than SCAN in locating values in inverted fields, and STEPSCAN (discussed next) and other commands are better tailored for operations associated with non-inverted fields. Therefore, you should examine STEPSCAN before you use SCAN and determine which of these two commands is most appropriate for your needs.

Like the FIND command, SCAN treats the entries in the field in a variable length fashion; that is, you can ask for several values to be present in a particular field, and the search will be satisfied if the values exist in any step. Sixteen fields (the fields in the coded step matrix) are available for use with the SCAN command:

LK	(Link)	TR	(Train)
SLK	(Sublink)	CH	(Channel)
CAUSE	(Cause)	DI	(Differ)
PSYS	(Primary system)	T	(Timing)
ISYS	(Interface system)	P	(Performance)
COMP	(Component)	D	(Detection)
VEND	(Vendor)	EFF	(Effect)
QUAN	(Quantity)	ICOMP	(Coarse component code)

Syntax

The syntax for SCAN is the same as that for FIND, STEPSCAN, VALUES, RELATE, LINK, JOIN, and TREND. Like all SCSS searching commands, SCAN uses options 2, 3, and 4 discussed on pp. 3-14 through 3-21.

You can limit the search to a given date range by including the terms

RANGE date1 date2

on the SCAN command line. Date1 is the beginning date of the desired range, and date2 is the ending date. Dates are given as mm/dd/yy (month/day/year).

If you want to use the same searching logic with SCAN that you used in your preceding search, you can enter

SCAN *

Example

```
*
FIND
+
<comp> hang
+
END
==>GROUP 1 HAS 39 LERs--OPERATION: "FIND"
*
SCAN
+
<tr> .GT. 1
+
END
==>GROUP 2 HAS 2 LERs OUT OF 39 SEARCHED--OPERATION: "SCAN"
            SECONDS: 0.89 (CPU)  19.02 (CLOCK)--RATIO:0.047
```

In this example, a FIND command was used to locate LERs containing the component code for pipe hangers (hang). These LERs were then scanned to find those involving two or more trains (<tr> .GT. 1).

STEPSCAN

Function

The STEPSCAN command is provided so that you can search an LER on a step-by-step basis, checking only the values within an individual step against the searching logic. Because screening at the step level is needed so frequently, STEPSCAN is one of the more useful and valuable SCSS commands.

STEPSCAN, like the SCAN command, can search both inverted and non-inverted fields (all the fields in the coded step matrix, as listed with the SCAN command). Like SCAN, STEPSCAN can be used with options 2, 3, and 4 discussed on pp. 3-14 through 3-21.

Step-level screening requires more care in specifying searching logic than the LER-level screening characterized by the FIND and SCAN commands because a field can have only one value within a step. (Therefore, asking for steps involving component codes for diesel and motor within the same step, for example, is not logical.)

STEPSCAN can be used by activating the entire database, using FIND ALL, then entering a STEPSCAN command line. However, STEPSCAN is slow if it is used on the entire database because it does not use the inverted fields. Consequently, you should use the FIND command on the inverted fields then use STEPSCAN on the resulting smaller group, as shown in the following example.

Syntax

The syntax for STEPSCAN is the same as that for FIND, SCAN, VALUES, RELATE, LINK, JOIN, and TREND. Review options 2, 3, and 4 discussed on pp. 3-14 through 3-21.

Often, you will want to perform a STEPSCAN search using the searching logic that you defined for the preceding FIND. In this case, simply enter

STEPSCAN *

This will enable you first to locate LERs that contain particular codes then to locate those LERs in which the codes occur within the same step.

You can also specify a date range using STEPSCAN by entering

STEPSCAN RANGE date1 date2

This will restrict the search to LERs with event dates that fall within the period beginning with date1 and ending with date2 (mm/dd/yy=month/day/year).

Example

```
*
FIND
+
<comp> dsl <psys> hw
+
END
==>GROUP 1 HAS 8 LERs--OPERATION: "FIND"

*
STEPSCAN *
THE FOLLOWING "STEP" DATA COLLECTIONS WILL BE CAPTURED:
     THE STEPS SATISFYING THE CURRENT SEARCHING LOGIC
==>GROUP 2 HAS 3 LERs OUT OF 8 SEARCHED--OPERATION:  "STEPSCAN"
     3 STEPS SATISFIED THE SEARCH STRATEGY
          SECONDS:  0.20 (CPU)   2.43 (CLOCK)--RATIO: 0.082
```

The FIND command looked for LERs that contained a diesel component and a high pressure core spray system anywhere in the LER (i.e., not necessarily in the same step). The STEPSCAN * command looked for LERs that contained the values in the same step, narrowing the active record group from eight to three.

TEXTSCAN

Function

The TEXTSCAN command, which searches abstracts, locates LER records that contain specific text strings.

One or two text strings (up to 64 characters in length) can be searched for at a time; however, the text strings must be input on separate lines. By default, a hit occurs if either string appears within an abstract. By specifying **BEFORE**, **AFTER**, or **AROUND**, you can request that both strings occur and also specify the order in which they must occur.

Syntax

TEXTSCAN

At the + prompt, enter

'text string1'
+
BEFORE
+
'text string2'
+
END

Note that text strings require single quotation marks (') as delimiters and can contain embedded blanks.

To specify an order in which the text strings must appear, enter **BEFORE** if the first text string should appear first, enter **AFTER** if the first text string should appear second, or enter **AROUND** if both text strings should appear, but in any order.

Example

```
*
FIND
+
<comp> hang
+
END
==>GROUP 1 HAS 39 LERs--OPERATION:  "FIND"
*
TEXTSCAN
+
'eccs'
+
AFTER
+
'test'
+
END
==>GROUP 2 HAS 1 LERs OUT OF 39 SEARCHED--OPERATION: "TEXTSCAN"
```

This example found one LER with pipe hangers in the component field and with tne text string 'eccs' (for emergency core cooling system) appearing after the text string 'test' in the abstract.

TABLE

Function

The TABLE command provides a means of decoding the codes into meaningful text and also goes one step further and allows for decoding the syntax as it will be interpreted by a specific searching or processing command. SCSS uses non-mnemonic terms that bear little resemblance to the item being coded, and the TABLE command prevents you from having to memorize the codes or work with the extensive list of codes. Whenever you issue any of the searching commands--for example, FIND, SCAN, STEPSCAN, or LINK--a syntax interface program interprets the commands and creates a logic table, which all of the commands use to perform their tasks. The TABLE command allows you to see this logic table, which will show you the logical relationships established in your commands. The TABLE command is especially helpful for users who are working with advanced commands like LINK.

Syntax

TABLE

This simple syntax will decode the codes and specify the logical connections.

TABLE command

You can also issue the TABLE command with the name of a specific command and receive more interpretation of how that command will use the syntax.

Example

```
*
FIND
+
<comp> (pno plo) pux <eff> tb
+
END
==>GROUP 1 HAS 46 LERs--OPERATION: "FIND"
*
TABLE FIND

-----------------------FIND-----------------------

    COMP = PNO--NON-LICENSED OPERATOR PERSONNEL
      OR = PLO--LICENSED OPERATOR PERSONNEL
AND COMP = PUX--OTHER/UNKNOWN UTILITY PERSONNEL
 AND EFF = TB--CONSEQUENTIAL OMISSION OF TASK, ANALYSIS,
           OR STEP
```

The table demonstrates that the entry **<comp> (pno plo) pux <eff> tb** requests LERs that have component codes of pno and pux or plo and pux and the effect code of tb.

VALUES

Function The VALUES command can be used to find steps and process them to locate unique codes for the 16 fields in the coded step matrix (see p. 3-29) and 6 LER-level fields, to count the usage of the codes, and to produce decoded tables of counts of code usage. The six LER-level fields are DOCKET, YEAR, TYPE (reactor type), NSSS (NSSS supplier), AE (architectural engineer), and SYMBOL (symbol for reactor operator). When you use VALUES, your screen will show a count of unique values and a table listing (1) the unique values appearing in your active record group, (2) the number of steps (percentages), and (3) a description (or translation) of the value codes. This is an extremely valuable command for analyzing LERs.

NOTE 1: The ICOMP field can be used in the logic to screen steps, but VALUES will not work to obtain counts for this field. See the SVALUES command discussion in Chap. 4 for a way to obtain this kind of analysis.

NOTE 2: You can use VALUES to count usage in the six LER-level fields--DOCKET, YEAR, TYPE, NSSS, AE, and SYMBOL--but you cannot count steps in these fields because they are not part of the coded step matrix.

Although VALUES can be used to search the database, this discussion tells you how to use VALUES to tabulate usage after you have developed your active record group using other commands. Advanced uses of VALUES are discussed in Chap. 4.

Like STEPSCAN and other SCSS commands, VALUES works at the step level and uses options 2, 3, and 4 discussed on pp. 3-14 through 3-21.

Syntax

VALUES * <field>

This syntax will produce a table of values and counts appearing in the active record group you have already established. The asterisk (*) entered after VALUES tells the computer to use the current logic. After the asterisk, enter the name of the field you would like the count performed on. You can enter more than one field.

As you can with other SCSS commands, you can limit the VALUES count to a given date range by entering

VALUES * RANGE date1 date2 <field>

The range must precede the field name(s) and the dates must be given in the form mm/dd/yy (month/day/year).

By default, the table produced by VALUES is sorted by count in descending order. You can also tell the computer to sort the codes alphabetically:

VALUES * KEYSORT field

By entering **KEYSORT** before the field name, you have told the computer to produce an alphabetical listing by code value.

If you do not wish to see values that occurred infrequently, you can reduce the amount of output that the computer will give you in one of two ways. If you are dealing with more than one field, it is easiest to issue one of these commands, receive the output, then issue another command for the next field.

VALUES * TOP n <field>

TOP tells the computer that you only want to see the values that occurred most frequently, and **n** tells the computer that you want the top n uses.

VALUES * OVER n <field>

The **OVER** option allows you to specify how many times a value must be used in order for it to be displayed on your screen. A value must be used more than n times before it is included in the list on your screen. For instance, if you specify OVER 5, your list will include only those values that occurred in more than five steps.

Example 1

```
*
FIND
+
<comp> hang
+
END
==>GROUP 1 HAS 39 LERs--OPERATION: "FIND"
*
VALUES * <psys>
```

This search found all the steps with a pipe hanger (hang) component (comp) then produced a table of all the primary systems (psys) in which the hangers are located.

Example output is shown on p. 2-31.

Example 2

```
*
FIND
+
<vend> W120
+
END
==>GROUP 1 HAS 724 LERs--OPERATION: "FIND"
*
VALUES * TOP 5 <comp>
```

In this example, the user wanted to know the components supplied by Westinghouse Electric Corporation (vend = W120). When the FIND command located 724 LERs, the user decided to limit the amount of output she received by asking the computer to report only the five values that occurred most frequently in the component field.

RELATE

Function

The RELATE command finds all unique pair combinations of codes for any two fields in the coded step matrix. The 11 fields in the coded step matrix and the 6 LER-level fields listed for use with the VALUES command are available for use with RELATE. See the SCSS Command/Field Cross Reference Chart on p. 3-2.

The output from RELATE is a two-dimensional table that gives the counts for all usages. Optionally, the output can be displayed in percentage units. The two-dimensional table tries to arrange the two fields with the codes for the first field listed in a horizontal line at the top of the table, and the codes for the second field listed vertically. However, it will automatically reverse this ordering if the arrangement produces a smaller table size. The relationship counts appear in the appropriate spaces on the grid created by this arrangement of the codes in the two fields. Tables are printed in pieces when there are too many codes in the first field to fit within a single table.

Like the VALUES command, RELATE has options to limit its output. The same **TOP** and **OVER** options are available to limit the output to the codes that are used most often or that have usages more than some lower limit. However, because there are two fields to contend with, this usage is more involved. Including **TOP** n in the RELATE command line limits the output to the n highest used codes in the <u>first</u> field. To limit the second field, the same words are used, but they must come after **TOP n** for the first field (i.e., **TOP n TOP m** limits the output to the n highest used codes in the first field and the m highest used codes in the second field). The same rules on order apply to the **OVER** n option, which limits the output to codes used more than n times. If you want all the codes for the first field, you can either reverse the order of the fields on the command line or specify options such as **TOP** 99999 or **OVER** 0, which would be guaranteed to capture all uses.

Syntax

RELATE * field1 field2

RELATE uses options 2, 3, and 4 discussed on pp. 3-14 through 3-21 to specify its searching logic. To use the same logic that you used to locate the LER group, use an asterisk (*) as the second term in the RELATE command line; then specify the two fields you are interested in.

Example

```
*
FIND
+
<comp> (plo pux)
+
END
==>GROUP 1 HAS 78 LERs--OPERATION: "FIND"
*
RELATE * <psys> <comp>
```

When this series of commands was issued, the computer scanned the 78 LERs in the active record group then reported that it found 4 unique values in the primary system field and 2 unique values in the component field for all steps with a licensed operator personnel (plo) and other/unknown utility personnel (pux) in the component field. It also displayed a grid that named the unique values and told how many times each value appeared in both fields.

Example output is shown on p. 2-33.

TREND

Function

The TREND command determines how certain types of events occur as a function of time and produces a two-dimensional matrix showing how all unique codes for a particular field correspond with the event dates of the LERs. TREND uses a step-level logic to locate the steps that are processed; i.e., this command works like STEPSCAN to locate steps. See the SCSS Command/Field Cross Reference Chart on p. 3-2.

The fields available for use with the TREND command are those available for use with the VALUES command. Like VALUES, TREND offers syntax options that allow you to limit the amount of output you receive.

Syntax

TREND * <field>

TREND uses options 2, 3, and 4, discussed on pp. 3-14 through 3-21, to specify its searching logic. You can also use the same logic that you used to locate the LER group by including an asterisk (*) as the second term in the TREND command line then specifying the field containing the values you would like related to event dates.

By default, the dates used by TREND are divided by quarter, beginning with January 1, 1981. However, you can elect to have TREND use monthly dates or you can specify a range of dates. The output is normally sorted by count on code usage in descending order, although an alphabetical sort can be selected by including the term **KEYSORT** on the command line.

TREND * <field> MONTH

By including **MONTH** on the command line, you have asked the computer to provide you with counts by month.

TREND * <field> RANGE date1 date2

Date1 is the beginning date of your range, and **date2** is the ending date. Dates must be entered as mm/dd/yy (month/day/year).

The TREND command also offers two options that can help avoid lengthy output.

TREND * <field> TOP n

The terms **TOP n**, where n is an integer, tell the computer to include only the codes with <u>total</u> counts that are the n largest in the requested field. (Note, however, that this is <u>not</u> the count by individual date.)

TREND * <field> OVER n

By entering **OVER n**, where n is an integer, you will eliminate processing for codes with a total count (for all event dates within the specified range) that is smaller than the integer n.

You can combine the TREND syntax options in the same command line. They can be entered in any order.

Example

```
*
FIND
+
<comp> (plo pux)
+
END
==>GROUP 1 HAS 78 LERs--OPERATION: "FIND"
*
TREND * <psys>
```

In the grid that resulted from this command, the computer displayed the four unique values that occurred on the horizontal axis, each quarter from January 1, 1981, to March 1, 1984, on the vertical axis, and the occurrence counts in the boxes.

Example output is shown on p. 2-34.

RECALL

Function

The RECALL command is one of a set of commands that manipulate the LER groups you have established during processing. (The group number is shown on the screen as ==>GROUP n, where n is the integer assigned by the computer to identify the group.)

RECALL is the simplest command for dealing with LER groups. It returns you to the LERs in a group that was found during an earlier step in the current sequence of searching commands.

Syntax

RECALL n

n is the number of the active group you want to return to.

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 591 LERs--OPERATION: "FIND"
*
LOCATE SYMBOL=tva
==>GROUP 2 HAS 61 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
*
RECALL 1
==>LER GROUP 3 CONTAINS THE 591 LERs FROM GROUP 1
```

After you have recalled a group, you can continue processing and the computer will use the group you have just recalled. For instance, if you next wanted to locate LERs for MEC facilities, you could enter

```
LOCATE SYMBOL=mec
```

and the computer would search the 591 LERs in group 3 for plants operated by Metropolitan Edison.

COMBINE

Function The COMBINE command combines two or more record groups from a processing series into a single group, eliminating duplicate LERs. COMBINE can also be used with permanently saved record groups (see SAVE).

Syntax

COMBINE n m

For n and m, substitute the numbers, names, or a combination of the record groups you want to combine. Note that more than two groups are allowed.

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 591 LERs--OPERATION: "FIND"
LOCATE SYMBOL=tva
==>  OUP 2 HAS 61 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
*
RECALL 1
==>LER GROUP 3 CONTAINS THE 591 LERs FROM GROUP 1
*
LOCATE SYMBOL=mec
==>GROUP 4 HAS 13 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
*
COMBINE 2 4
==>GROUP 5 HAS 74 LERs--OPERATION: "COMBINE"
```

MATCH

Function

The MATCH command identifies those LERs common to groups created by different search strategies. MATCH is helpful if you have located groups of LERs using different search strategies and want to determine any areas of overlap. MATCH can also be used with permanently saved recor' groups (see SAVE).

Syntax

```
*
MATCH n m etc.
```

For n and m, substitute the number, name, or a combination of the record groups you want to check for duplicates. Note that more than two groups are allowed.

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 591 LERs--OPERATION: "FIND"
*
LOCATE NSSS=ge
==>GROUP 2 HAS 245 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
*
RECALL 1
==>LER GROUP 3 CONTAINS THE 591 LERs FROM GROUP 1
LOCATE SYMBOL=tva
==>GROUP 4 HAS 61 LERs OUT OF 591 SEARCHED--OPERATION: "LOCATE"
*
MATCH 2 4
==>GROUP 5 HAS 36 LERs--OPERATION: "MATCH"
```

Note that the above example could have asked for both TVA and GE reactors in the first LOCATE command.

```
LOCATE NSSS=ge SYMBOL=tva
```

NOMATCH

Function The NOMATCH command identifies LERs that occur only in one of two LER groups. NOMATCH can use permanently saved record groups (see SAVE).

NOTE: NOMATCH only processes two groups at a time.

Syntax

NOMATCH n m

For n and m, substitute the numbers or the names of the record groups you want checked.

Example

```
*
NOMATCH 2 5
==>GROUP 6 HAS 209 LERs--OPERATION: "NOMATCH"
```

This example continues where the example for the MATCH command ended. The NOMATCH on groups 2 (LERs involving diesel generators at GE-manufactured reactors) and 5 (LERs involving diesel generators at GE-manufactured reactors operated by TVA) identifies those LERs involving diesel generators at GE reactors other than those operated by TVA.

If you do a LOCATE on group 6 for SYMBOL=tva, you will not get any LERs; LOCATE TYPE=bwr gives you all 209 LERs. The NOMATCH, though, found LERs involving diesels at other GE reactors besides those operated by TVA.

NOTFOUND

Function Whenever you screen a group of LERs to locate those that satisfy some searching criteria, two collections result: (1) those LERs that satisfy the search and (2) those that do not. The NOTFOUND command activates the second collection.

Syntax

```
NOTFOUND
```

Example

```
*
FIND
+
<comp> dsl
+
END
==>GROUP 1 HAS 591 LERs--OPERATION: "FIND"

*
NOTFOUND
==>GROUP 2 HAS 12334 LERs--OPERATION: "NOTFOUND"
```

In this example, group 2 is made up of LERs that do not contain the component code for diesel motors.

KEYSORT

Function After a sequence of one or more searching operations, the record keys in the active record group may or may not be ordered according to any recognizable pattern (e.g., by docket, then by year, and finally by LER number). The KEYSORT command forces the ordering to be by docket, year, and LER number.

Syntax

KEYSORT

As an example, assume you have asked for LERs containing the coarse component codes for motors, personnel, and valves (ICOMP=250, 270, and 350, respectively). A listing of the keys by the DISPLAY command shows an unordered set of keys.

Example

```
*
FIND
+
<icomp> 250 270 350
+
END
==>GROUP 1 HAS 33 LERs--OPERATION: "FIND"
*
DISPLAY
```

```
 1 269/81-001     2 261/81-032     3 271/81-023     4 281/81-026
 5 281/81-034     6 315/81-004     7 325/81-042     8 327/81-145
 9 344/81-026    10 346/81-037    11 364/81-053    12 369/81-173
13 369/81-179    14 369/81-183    15 206/82-015    16 219/82-044
17 247/82-017    18 281/82-054    19 311/82-087    20 315/82-041
21 315/82-101    22 336/82-013    23 361/82-103    24 368/82-021
25 302/83-044    26 312/83-022    27 334/83-010    28 387/83-056
29 387/83-140    30 409/83-007    31 416/83-168    32 416/83-049
33 305/83-036
```

This shows an unordered set of keys. After a KEYSORT, the keys are ordered, thereby making it easier to deal with listings of all of the LERs.

```
*
KEYSORT
*
DISPLAY
```

```
 1 206/82-015     2 219/82-044     3 247/82-017     4 261/81-032
 5 269/81-001     6 271/81-023     7 281/81-026     8 281/81-034
 9 281/82-054    10 302/83-044    11 305/83-036    12 311/82-087
13 312/83-022    14 315/81-004    15 315/82-041    16 315/82-101
17 325/81-042    18 327/81-145    19 334/83-010    20 336/82-013
21 344/81-026    22 346/81-037    23 361/82-103    24 364/81-053
25 368/82-021    26 369/81-173    27 369/81-179    28 369/81-183
29 387/83-056    30 387/83-140    31 409/83-007    32 416/83-049
33 416/83-168
```

LINK

Function

Two fields in the coded step matrix, link (LK) and sublink (SLK), specify relationships between steps in an event. Their function is essentially identical except that the sublink field is used whenever a particular step results from a combination of two or more earlier steps.

The linkages are important because they contain information about the causes that lead to a particular kind of step or the resulting effects of a particular step. For example, if you are analyzing a motor failure, it might be essential to know what kinds of steps preceded or followed that motor failure and were also directly linked to it.

The LINK command allows you to make searches concerning the linkages between steps. As such, it provides a powerful analysis tool that begins to make use of the complexities and generalities encoded in the coded step matrix. Any of the fields in the coded step matrix can be used with LINK.

The LINK command looks for linkages between cases. The first case is always the target case, and the command can look for linkages occurring either before or after steps that satisfy the case 1 logic to steps that satisfy logic for any of the later cases. By default, the command looks for LERs in which case 1 links to later steps that satisfy case 2, case 3, etc. (i.e., it looks for case 1 leading to a later case). Case 1 must occur first.

LINK uses options 2, 3, and 4, discussed on pp. 3-14 through 3-21, to specify its searching logic.

Syntax

LINK

At the + prompt, enter one of the cases you are interested in.

<field> value etc.

A second + prompt will appear; enter your second case.

<field> value etc.

The LINK command must always be used with at least two cases, but can be used with more than two cases. In this format, the link will look for the first case being linked to later steps that satisfy cases 2 or 3 or 4, etc.

By adding **BEFORE** on the LINK command line, you can make the computer look for linkages backwards; that is, case 1 must occur after later cases.

```
LINK BEFORE
+
<field> value
+
<field> value
.
.
.
+
END
```

In this form, the LINK will look for the first case being linked to earlier steps that match case 2 or 3 or 4, etc.

If you are interested in linkages occurring in EITHER direction, before or after, you can specify **EITHER** on the command line.

```
LINK EITHER
+
<field> value etc.
+
<field> value etc.
+
END
```

This form looks for linkages between case 1 and case 2, when case 2 can be either before or after case 1.

Example 1

```
*
FIND
+
<icomp> 250 270
+
END
==>GROUP 1 HAS 92 LERs--OPERATION: "FIND"
*
LINK
+
<icomp> 270
+
<icomp> 250
+
END
THE "LINK" WILL LOOK FOR THE FIRST "CASE" BEING LINKED TO LATER
STEPS WHICH SATISFY CASES 2,3,4.....
==>GROUP 2 HAS 79 LERs OUT OF 92 SEARCHED--OPERATION: "LINK"
                SECONDS: 1.58 (CPU) 18.25 (CLOCK)--RATIO:0.087
```

In this example, the user was interested in LERs involving personnel actions that were linked to motor failures. In the database, motor component codes are grouped under the coarse component code 250 (ICOMP=250) and personnel codes are grouped under ICOMP=270. As with any searching or processing command other than FIND, it is prudent to reduce the number of LERs that will be processed. The simplest way, for this case, is to ask for LERs containing both ICOMP values--250 and 270--and to then execute the LINK command.

The screen shown in Fig. 3-3 is one of the 79 LERs located by the previous LINK search.

The personnel step (step 1, COMP=PUX) is not linked directly to the motor step (step 4, COMP=MOT), but has a link through steps 2 and 3.

FIGURE 3-3. EXAMPLE OF LINK

```
*
LIST 5

FORM     5                    LER SCSS DATA                              04-25-84
**********************************************************************************
DOCKET   YEAR   LER NUMBER   REVISION   DCS NUMBER    NSIC      EVENT DATE
  261    1981       032          0     8201130297   171626     12- 5-1981

DOCKET:261  ROBINSON 2                    TYPE:PWR
                  REGION:  2             VENDOR:WEST
ARCHITECTURAL ENGINEER: EBASCO
     FACILITY OPERATOR: CAROLINA POWER & LIGHT CO.
                  SYMBOL: AHV

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0       SH   PO       PUX         Z          A A T  F  UB
  2   1       RC   BK       ZVZ         1    1     1 A TR F  AE
  3   2       FB   BK       PMPB C300   1    1     1 A T  F  HA
  4   3       HA   BK       MOT         1    1     1 A T  F  IH
  5            XX                                    E XX    YC
  6            YY                                    N N     YC

ABSTRACT
 BORIC ACID (BA) TRANSFER PUMP 'A' TRIPPED AND WAS DECLARED INOPERABLE.
 THE PUMP TRIP WAS CAUSED BY AN ERROR IN VALVE LINEUP PROCEDURE
 OP-28-3-A WHICH SPECIFIED VALVE CVC-338 ('A' BA TRANSFER PUMP SUCTION)
 BE CLOSED CAUSING CAVITATION OF THE PUMP.  THIS EVENT RESULTED IN
 OPERATION IN A DEGRADED MODE PERMITTED BY TECH. SPEC. BA TRANSFER PUMP
 'A' MOTOR WAS FOUND TO HAVE AN OPEN WINDING AND WAS REPLACED.  THE
 PUMP WAS TESTED AND DECLARED OPERABLE.  THE DEFECTIVE PROCEDURE, WAS
 REVISED TO INCLUDE THE CORRECT POSITION OF VALVE CVC-338.
```

Example 2 The screen shown in Fig. 3-4 is one of the four LERs that satisfied the preceding search criteria when **BEFORE** was added to the command line.

Note that step 1 (COMP=MOT) links through steps 2 and 3 to step 4 (COMP=PZ, unknown personnel).

FIGURE 3-4. EXAMPLE OF LINK USING BEFORE OPTION

```
*
LINK BEFORE
+
<icomp> 270
+
<icomp> 250
+
END
THE "LINK" WILL LOOK FOR THE FIRST "CASE" BEING LINKED
TO EARLIER STEPS THAT MATCH CASES 2, 3, 4, ETC.

==>GROUP 2 HAS 4 LERs OUT OF 92 SEARCHED--OPERATION: "LINK"
     SECONDS:  1.52 (CPU)  16.38 (CLOCK)--RATIO:  0.093
*
LIST 2
FORM     2                  LER SCSS DATA                          04-25-84
*****************************************************************************
DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER    NSIC      EVENT DATE
  369   1981       173          0  8112140306  171144     10-31-1981

DOCKET:369  MCGUIRE 1                    TYPE:PWR
                 REGION:  2             VENDOR:WEST
ARCHITECTURAL ENGINEER: DUKE
     FACILITY OPERATOR: DUKE POWER CO.
                 SYMBOL: APY

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
   1  0        ZZ   FI        MOT        1     1     1 A T  K  IF
   2  1        RC   FI        VOB        1     1     1 A TR F  KF
   3  2        RC   FI        ISVZ       1     1     1 A M  I  KC
   4  3        SX   PM        PUX        Z           A A T  K  UB
   5  4        RC   FI        VOA        1     1     1 A T  F  AE
   6  5        RC   FI   BA   ISVZ       1     1     1 A T  F  KA
   7  6        RS   FI   BA   1XZ        1     1     1 A T  F  BI
   8           XX                                     K XX     YC
   9           YY                                     N N      YC


ABSTRACT
 DURING PERFORMANCE OF MAINTENANCE ACTIVITIES, STEAM GENERATOR 1B MAIN
 FEEDWATER TO AUXILIARY FEEDWATER NOZZLE ISOLATION VALVE WAS
 INADVERTENTLY OPENED.  AS A RESULT OF PERSONNEL ERROR, THE VALVE'S
 HANDWHEEL WAS TURNED IN THE WRONG DIRECTION, LIFTING THE DISC OFF THE
 SEAT ENOUGH TO ALLOW SYSTEM PRESSURE (APPROXIMATELY 1140 PSIG) TO
 COMPLETELY OPEN THE VALVE.  THE VALVE WAS JACKED CLOSED AND GAGGED.
 THE VALVE'S ACTUATOR WAS REPAIRED, REINSTALLED AND TESTED AND THE
 VALVE DECLARED OPERABLE.
```

REF

Function Many LERs list reference LERs that are associated with a given LER. After you locate an LER group, you can access the reference LERs by issuing the REF command.

NOTE: The SCSS database currently contains LERs from 1981 through the present, and these LERs may reference previous LERs (pre-1981) that are not actually in the database.

Syntax

REF LIST

REF is a one-line command consisting of the term REF, which finds the reference LERs and puts them in the active record group, optionally followed by the term LIST, if you want to list the source LERs for the reference LERs.

Example

```
*
FIND
+
<comp> hang
+
END
===>GROUP 1 HAS 39 LERs--OPERATION: "FIND"
*
REF LIST
```

LER	REFERENCE LERS			
23781018	23777010	23779016	23780016	
23781057	23770035			
23781061	23781057			
36681104	36681120			
33182014	33177008	33177029	33179025	33182008
36182032	36182027			
36983018	36983016			
36983023	36983016	36983018		

```
==>GROUP 2 HAS 13 LERs OUR OF 39 SEARCHED--OPERATION: "REF"
          SECONDS:   0.68 (CPU)    6.78 (CLOCK)--RATIO:0.100
```

This example located LERs dealing with pipe hangers (comp=hang) then captured the reference LERs. The example used the LIST option to list the source LERs from the original active group.

BUILD

Function In some cases, you will know exactly which LERs you want in your active record group, independent of any searching strategy. Generally, these special groups will consist of a few LERs. You can use the BUILD command to specify the LERs that will be included in the active record group. Note that BUILD will produce an error message if the LERs you request are not for valid records.

Syntax

BUILD dddyynnn etc.

The BUILD command is a one-line command in which you enter the term BUILD followed by up to seven 8-digit LER keys. If you want more LERs in the active record group, you can use the BUILD command repeatedly and then combine the active record groups into one group using the COMBINE command. The record keys are specified as dddyynnn, where ddd is the docket number, yy is the year, and nnn is the LER number.

Example 1

```
*
BUILD 02982011 41683013

LER GROUP 3 ************************************************

THE "BUILD" HAS CREATED AN ACTIVE LIST OF 2 LERs

*
DISPLAY

     1 029/82-011    2 416/83-013
```

Example 2

```
*
BUILD 99980999

THE DOCKET CODE:999 IN POSITION:  2 SEEMS STRANGE
THE RECORD KEY WAS:99980999

LER GROUP 5 ************************************************

THE "BUILD" HAS CREATED AN ACTIVE LIST OF 0 LERs
```

In this example, 999/80/999, an invalid number, was entered as a key.

REMOVE

Function The REMOVE command allows you to remove specific LERs from an active record group.

Syntax

REMOVE n

where n is the sequential identifier of the LER, or

REMOVE dddyynnn

where ddd is the docket number, yy is the year, and nnn is the LER number

REMOVE is a one-line command with the term REMOVE followed by either the sequential identifier of the keys to be removed from the active record group or the keys themselves.

Example

```
*
FIND
+
<comp> hang
+
END
==>GROUP  1 HAS  39 LERs--OPERATION:  "FIND"
*
DISPLAY
 1 237/81-018     2 237/81-057     3 237/81-061     4 265/81-025
 5 289/81-007     6 312/81-029     7 317/81-029     8 325/81-034
 9 327/81-069    10 327/81-072    11 334/81-001    12 336/81-008
13 336/81-032    14 338/81-035    15 366/81-104    16 366/81-120
17 368/81-037    18 369/81-025    19 250/82-007    20 302/82-044
21 331/82-014    22 331/82-022    23 346/82-033    24 361/82-032
25 366/82-018    26 369/82-016    27 373/82-121    28 395/82-032
29 255/83-065    30 272/83-020    31 309/83-002    32 369/83-016
33 369/83-018    34 369/83-023    35 387/83-034    36 324/83-090
37 327/83-162    38 336/83-031    39 395/83-138
*
KEYSORT
*
DISPLAY
 1 237/81-018     2 237/81-057     3 237/81-061     4 250/82-007
 5 255/83-065     6 265/81-025     7 272/83-020     8 289/81-007
 9 302/82-044    10 309/83-002    11 312/81-029    12 317/81-029
13 324/83-090    14 325/81-034    15 327/81-069    16 327/81-072
17 327/83-162    18 331/82-014    19 331/82-022    20 334/81-001
21 336/81-008    22 336/81-032    23 336/83-031    24 338/81-035
25 346/82-033    26 361/82-032    27 366/81-104    28 366/81-120
29 366/82-018    30 368/81-037    31 369/81-025    32 369/82-016
33 369/83-016    34 369/83-018    35 369/83-023    36 373/82-121
37 387/83-034    38 395/82-032    39 395/83-138
*
REMOVE 27 28 29
LER GROUP 4 ************************************************
THE ACTIVE LIST CONTAINS  36 LERs AFTER THIS "REMOVE"
```

SAVE

Function

At any point in constructing a sequence of searching commands, you can save the logic and the resulting active record group by issuing the SAVE command. To retrieve a saved file, use ACTIVE and EXEC.

One particularly valuable feature of SCSS is the ability to store the logic for a sequence of searching commands in a special file, which can subsequently be referenced and reexecuted, thereby saving time and helping to ensure the new search is exactly what you want. In addition to searching logic, the special file contains the keys for all the LERs that satisfied the search when the command sequence was constructed.

Whenever you save searching logic, you will give it a name made up of one to eight characters, one of which must be alphabetic (e.g., MOTORS, PUMPS, A123, and 12A3 are all valid names, whereas 123 is not). These names can be used with any command that deals with active record groups, in combination with other saved record groups, or with numbered record groups out of the current searching sequence.

Most SCSS commands can be included in a saved searching sequence. These include all of the major searching and processing commands, but do not include commands that delete or activate saved record groups. The following are the commands that will be saved.

BUILD	MATCH	SRELATE
COMBINE	NOMATCH	STEPLIST
DISPLAY	NOTFOUND	STEPSCAN
FIND	RECALL	STEPSORT
JOIN	REF	SVALUES
KEYSORT	REMOVE	TEXTSCAN
LINK	SAVE	TREND
LIST	SCAN	VALUES
LOCATE		XFIND

All SCSS commands can be used while constructing a sequence of commands to save; however, only the commands listed above will be stored.

SAVE is a one-line command consisting of the term SAVE followed by the filename under which to catalog the search. In the event that a previous SAVE used the name you want for the new package, the computer will indicate that the filename has already been used. You can include an R as the third term to overwrite the older set.

You will want to use discretion in saving searching logic, especially when your search located large numbers of LERs, because of space limitations within the database for retaining information of this type. As an SCSS user, you have been assigned your own block of storage, which

is sufficient to accommodate 10 to 20 typical searching strategies, along with other data that may be retained as a result of other commands. If you attempt to overflow your work space, an error message will result. This error message may terminate your session and require maintenance activity by the ORNL personnel responsible for the SCSS software in order to restore your ability to save data.

Syntax

SAVE filename

ACTIVE

Function To activate a set of searching logic saved by the SAVE command, use the ACTIVE command.

A group that has been activated can be used in two ways.

1. You may wish to continue processing the group of LERs that had resulted from the searching strategy when it was saved; in this case, simply continue.
2. You can reexecute the searching/processing strategy by issuing the term EXEC as a separate command, in which case the software sequentially executes the commands included in the strategy. If the contents of the database are different from those when the command strategy was constructed and saved, the new record group and strategy can be saved under a different (or the same) name. (Another useful process is to perform a NOMATCH between the new record group and the name under which the previously found group of LERs was stored to identify those new LERs that were located by the strategy.)

Syntax

ACTIVE filename

ACTIVE is a one-line command, consisting of the term ACTIVE followed by the filename of the saved searching strategy to be made active.

DELETE

Function The DELETE command can be used to delete saved files that you no longer want; it can also be used to delete LERs associated with unwanted codes when using commands such as the FIND HELP command (as illustrated on p. 2-8 of the tutorial).

NOTE: The DELETE command does not physically delete records from the database; rather, it marks the records as "deleted." Subsequent saves do not pick up this space, which is recovered by special database maintenance activities performed by ORNL personnel responsible for the SCSS database.

Syntax

DELETE filename

DELETE is a one-line command, consisting of the term DELETE followed by the filename of the saved files to be deleted.

DIRECTORY

Function The DIRECTORY command is provided to produce a directory of all the saved searching strategies currently available in your work area.

Syntax

DIRECTORY

Example

```
*
DIRECTORY

     8 NAME-LISTS HAVE BEEN LOCATED
LIST-NAME     NUMBER OF LERS

AGAST              69     COMMANDS: FIND     ABSTRACT DISPLAY RECALL
                                    ABSTRACT COMBINE
BGAST              26     COMMANDS: FIND     ABSTRACT DISPLAY
BOLT               61     COMMANDS: FIND     STEPSCAN DISPLAY LIST
                                    STEPSCAN
FOIA              278     COMMANDS: FIND     COMBINE
PWRDIESE          347     COMMANDS: FIND     LOCATE VALUES
PWRDSL            341     COMMANDS: FIND     LOCATE VALUES
RELAY              44     COMMANDS: FIND     ABSTRACT
TEXT               26     COMMANDS: FIND     ABSTRACT COMBINE
```

EXEC

Function The EXEC command is used to reexecute the sequence of searching commands activated by the ACTIVE command.

Syntax

EXEC

Example After issuing an EXEC command, you may want to issue a NOMATCH command to check the final record group against the saved record group to identify the new LERs found by the searching strategy.

```
*
EXEC

THE "SEARCH" WILL CONSIST OF   3 COMMANDS

==============
COMMAND  1-->FIND
==============
==>GROUP  1 HAS  599 LERs--OPERATION:  "FIND"
==============
COMMAND  2-->LOCATE  TYPE    PWR
==============
==>GROUP 2 HAS 347 LERs OUT OF 599 SEARCHED--OPERATION: "LOCATE"
==============
COMMAND  3-->VALUES  *     TOP   10    CAUSE
==============
THE ACTIVE LIST OF 347 LERs HAS 15 UNIQUE VALUES IN THE CAUSE FIELD
FOR STEPS THAT MATCH THE CURRENT LOGIC TABLE

KEY VALUE    NUMBER OF STEPS          DESCRIPTION
    RC        358 (83.5%)     RESULTANT COMPONENT FAULT
    ZZ         34 ( 7.8%)     UNKNOWN
    AK         21 ( 4.8%)     ANTICIPATORY MAINTENANCE
    XX          7 ( 1.6%)     OTHER
    PY          3 ( 0.7%)     OTHER POSTULATED EVENT
    CI          2 ( 0.5%)     OVERSPEED
    PB          1 ( 0.2%)     SMOKING/BURNING
    KD          1 ( 0.2%)     FAILURE TO CLOSE
    NR          1 ( 0.2%)     LOW AMBIENT TEMPERATURE
    NT          1 ( 0.2%)     WATER SPRAY/CASCADE/FLOOD
             SECONDS:         4.20 (CUP) 72.24 (CLOCK)--RATIO:0.058
```

STOP

Function You can enter the STOP command whenever you are at the command entry level (*). Using STOP will both terminate your session and ensure that any items you marked for permanent retention (see SAVE) are handled properly.

The computer will prompt to see if you want to continue the session. A "yes" response will return you to the command entry level; any other response will terminate the session.

Syntax

STOP

TABLE 3-1.
SCSS COMMANDS AND ALLOWED ABBREVIATIONS

Command	Abbreviation	Command	Abbreviation
ACTIVE	AC	NOMATCH	NM
AUTH		NOTFOUND	NF
BUILD		OPTIONS	
COMBINE	CO	RECALL	
DEFAULT		REF	
DEFINE		RELATE	
DELETE	DE	REMOVE	
DIRECTORY		SAVE	SA
DISPLAY	DI	SCAN	SC
DOCKET		SINVENT	SI
END		SRELATE	SR
EXEC	EX	STEPLIST	SL
FIND	FI or F	STEPSCAN	ST
HELP		STEPSORT	SS
INVENT		STOP	
INVERTED		SVALUES	SV
JOIN	JO	TABLE	TA
KEYSORT		TEXTSCAN	TS
LINK	LK	TIME	
LIST	LI or L	TREND	
LOCATE		VALUES	
MATCH	MA	XFIND	

4. ADVANCED COMMANDS

4. ADVANCED COMMANDS

4. ADVANCED COMMANDS

This chapter discusses several SCSS features that will be used infrequently by most people. The commands presented here are not necessarily any more complicated or time consuming than those discussed in Chap. 3; rather, they are features directed at a limited audience (for example, commands that deal with an inventory database used to track the LERs in SCSS).

This chapter also discusses how to process the captured step data collections that can be produced by the STEPSCAN command, commands that use the link and sublink fields in their searching strategies (the LINK and JOIN commands), an option to produce flow charts with the LIST command, and options to change the hierarchy the system uses to look for records.

PROCESSING CAPTURED STEP DATA COLLECTIONS

Three classes of steps can be captured by the STEPSCAN command.

1. Those steps that satisfy the searching logic given to the STEPSCAN command--referred to as the * collection.
2. Those steps that occur <u>before</u> the steps that satisfy the STEPSCAN logic and that link to these steps through the link or sublink fields--referred to as the BEFORE collection.
3. Those steps that occur <u>after</u> the steps that satisfy the STEPSCAN logic and that link to these steps through the link or sublink fields--referred to as the AFTER collection.

By default, step collection * (number 1 above) is <u>always</u> captured, whereas the BEFORE and AFTER collections must be selected when the STEPSCAN command is executed.

Four SCSS commands process the captured collections. These are:

1. the STEPSORT command, which will sort the collections according to any of the fields contained in the captured data collections;
2. the STEPLIST command for listing the data in the collections;
3. the SVALUES command for analyzing the data in the captured collection to determine all unique code values used in a particular field; and
4. the SRELATE command for analyzing the data in a captured collection to determine all unique pairs of codes for any two fields in the collection.

The output from the SVALUES and SRELATE commands is identical to that from the VALUES and RELATE commands, which perform analogous functions without actually capturing step data.

The fields available for use with the STEPSORT, SVALUES, and SRELATE commands are listed below.

CAUSE	(Cause)	LK	(Link)
CH	(Channel)	NUMBER	(LER number)
COMP	(Component)	P	(Performance)
D	(Detection)	PSYS	(Primary system)
DI	(Differ)	QUAN	(Quantity)
DOCKET	(Docket number)	REV	(Revision number of the LER)
EFF	(Effect)	SLK	(Sublink)
ICOMP[a]	(Coarse component code)	STEP	(Step number from the coded matrix)
ISEQ1[b]	(Sequence number in the event)	T	(Timing)
ISEQ2[b]	(Sequence type)	TR	(Train)
ISEQ3[b]	(Step type)	VEND	(Component vendor)
ISYS	(Interfacing system)	YEAR	(Year)

[a]ICOMP is available if it was assigned in the STEPSCAN command (see p. 4-3).

[b]ISEQ1, ISEQ2, and ISEQ3 are available if they were assigned by the STEPSCAN command (see p. 4-3).

OPTIONS FOR THE STEPSCAN COMMAND

Chapter 3 discussed the fundamental capabilities of the STEPSCAN command. This chapter discusses several additional options. All options discussed in this chapter are specified in the STEPSCAN command line, but the order in which the options are selected is arbitrary.

Option 1: Assigning New Fields to the Coded Step Matrix

Function Certain classes of data can be captured into special temporary files for subsequent processing. These files consist of the 16 fields included in the coded step matrix. In addition, the docket code, the year, the LER number, the revision, and the step numbers are included to allow for identifying the steps in subsequent processing.

Coarse component codes (ICOMP) are not included in the SCSS coded step matrix and are not automatically included in the captured step data collections. However, this can be accomplished by including the term ICOMP in the STEPSCAN command line, after which ICOMP is a valid field for any of the commands discussed in the remainder of this chapter.

Three other fields can also be assigned to the captured step data files by including the term ISEQ in the STEPSCAN command line. These are:

1. ISEQ1--the number of the sequence. It is assigned to distinguish between individual sequences for those events that include more than one sequence.
2. ISEQ2--the sequence type. There are three fundamental sequence types (basic, sublink, and crosslink) that are identifiable based upon the interrelationships of the individual steps as they form a given sequence.
3. ISEQ3--the step type. Each step is identifiable as an initiating, terminating, combining, etc., step depending on how it is linked with any predecessor step(s) and/or any successor step(s).

Only the more advanced user should be concerned with these fields in further analyzing LERs.

Any of the commands discussed in the remainder of this chapter can use these three fields when they are preassigned with the ISEQ option in the STEPSCAN command.

Option 2: Counting Hits

Function

The STEPSCAN command contains an option for displaying the number of steps that meet the searching criteria on a particular LER. This option is selected by including the term **COUNT** on the STEPSCAN command line. If the searching logic includes more than one case (type of step), the number of times each case is satisfied by an LER can be displayed, if the terms **CASE** and **COUNT** are included in the STEPSCAN command line.

Other options allow the rejection of an LER from the active record group if the searching logic is satisfied fewer than some specified number of times. Including the terms **CUT** n on the STEPSCAN command line, where n is the number of hits, rejects an LER that does not have n or more steps satisfying the searching logic. The terms **CASECUT n** perform an analogous function except that, in this case, at least one case (type of step) must occur n or more times before the LER will be accepted.

Syntax

STEPSCAN * COUNT CASE

The asterisk (*) in the command line tells the computer to use logic from the preceding operation. **COUNT** tells the computer to display the number of steps that meet the searching criteria on a particular LER. **CASE** tells the computer to display the count for the number of times each case is satisfied.

STEPSCAN * CUT n

By substituting an integer for n, you can tell the computer not to report any LER that does not satisfy the searching logic at least n times.

STEPSCAN * CASECUT n

By substituting an integer for n, you can tell the computer not to report any LER that does not have at least one case occurring n or more times.

Example

In the example shown in Fig. 4-1, a FIND located the pertinent LERs, then STEPSCAN counted the number of steps that referenced motors and diesel engines in two separate cases. The next STEPSCAN eliminated all LERs with fewer than three steps for diesel engines and motors and gave a count, by case, of LERs that satisfied the screening. Another STEPSCAN rejected all LERs without one case being satisfied three or more times.

FIGURE 4-1. EXAMPLE OF STEPSCAN, OPTION 2

```
*
FIND
+
<comp> mot dsl
+
END
==>GROUP  1 HAS  46 LERs--OPERATION:  "FIND"
*
STEPSCAN
+
<comp> mot
+
<comp> dsl
+
END
==>GROUP 2 HAS 46 LERs OUT OF 46 SEARCHED--OPERATION: "STEPSCAN"
    104 STEPS SATISFIED THE SEARCH STRATEGY
CASE   1 WAS SATISFIED   47 TIMES
CASE   2 WAS SATISFIED   57 TIMES
           SECONDS:    1.12 (CPU)    11.75 (CLOCK)--RATIO:0.095

*
STEPSCAN * CUT 3 COUNT CASE
DOCKET-YR-NUMBER: 23781027 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
DOCKET-YR-NUMBER: 24781018 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    2 TIMES
                     CASE:  2 WAS SATISFIED    1 TIMES
DOCKET-YR-NUMBER: 31181050 FITS THE SEARCH WITH    5 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    4 TIMES
DOCKET-YR-NUMBER: 36681094 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
DOCKET-YR-NUMBER: 24982045 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
DOCKET-YR-NUMBER: 27282012 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
DOCKET-YR-NUMBER: 27282035 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
DOCKET-YR-NUMBER: 41682156 FITS THE SEARCH WITH    3 STEPS <---
                     CASE:  1 WAS SATISFIED    1 TIMES
                     CASE:  2 WAS SATISFIED    2 TIMES
```

FIGURE 4-1, Continued

```
     DOCKET-YR-NUMBER: 24983005 FITS THE SEARCH WITH    3 STEPS <---
                          CASE:  1 WAS SATISFIED    1 TIMES
                          CASE:  2 WAS SATISFIED    2 TIMES
     DOCKET-YR-NUMBER: 38983001 FITS THE SEARCH WITH    3 STEPS <---
                          CASE:  1 WAS SATISFIED    1 TIMES
                          CASE:  2 WAS SATISFIED    2 TIMES

==>GROUP  3 HAS    10 LERs OUT OF    46 SEARCHED--OPERATION: "STEPSCAN"
     32 STEPS SATISFIED THE SEARCH STRATEGY
CASE   1 WAS SATISFIED    47 TIMES
CASE   2 WAS SATISFIED    57 TIMES
               SECONDS:      1.14 (CPU)     54.24 (CLOCK)--RATIO:0.021

*
STEPSCAN * CASECUT 3
==>GROUP  4 HAS     1 LERs OUT OF    10 SEARCHED--OPERATION: "STEPSCAN"
      5 STEPS SATISFIED THE SEARCH STRATEGY
CASE   1 WAS SATISFIED    11 TIMES
CASE   2 WAS SATISFIED    21 TIMES
               SECONDS:      0.47 (CPU)      5.04 (CLOCK)--RATIO:0.093

*
DISPLAY
  1  311/81-050
*
OPTIONS LIST MATRIX

FORM     1                LER SCSS DATA                          05-03-84
*************************************************************************
DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER   NSIC     EVENT DATE
  311   1981      050         1     8207080364  175161     6-25-1981

DOCKET:311  SALEM 2                    TYPE:PWR
                 REGION:  1          VENDOR:WEST
ARCHITECTURAL ENGINEER: PSE&G
     FACILITY OPERATOR: PUBLIC SERVICE ELECTRIC & GAS CO.
                 SYMBOL: AQB

REFERENCE LERS:
    1 311/81-002
```

FIGURE 4-1, Continued

STEP	LK	SLK	CAUSE	PSYS	ISYS	COMP	VEND	QUAN	TR	CH	DI	T	P	D	EFF
1	0		SZ	PI		PZ		Z			A	M	T	K	UB
2	1		RC	CK		CBL		M	1		1	M	TR	K	AE
3	2	A	RC	CK		SOL	A499	1	1		1	M	TR	K	KF
4	0	A	ZZ	CK		SOL	A499	1	1		2	M	T	K	KF
5	0	B	ZZ	CK		MOT		1	1		1	A	T	I	KF
6	A	B	RC	EH		TCHG		1	1		1	A	T	I	KF
7	6	C													YC
8	B		RC	EH		DSL		1	1		1	A	T	I	KF
9	8		RS	EH		1X2		1	1		1	A	T	I	YB
10				XX								L	XX		YC
11				YY								N	N		YC
12	0	C	ZZ	CI		MEI		1	1		1	A	T	K	PE
13	C		RC	EH		DSL		1	1		1	A	TR	I	KF
14	13		RS	EH		1XZ		1	1		1	A	T	I	YB
15				XX								E	XX		YC
16				YY								N	N		YC
17	6		RC	EH		DSL		1	1		1	A	TR	I	KF
18	17		RS	EH		1XZ		1	1		1	A	T	I	YB
19				XX								C	XX		YC
20				YY								N	N		YC
21	6		RC	EH		DSL		1	1		1	A	TR	I	KF
22	21		RS	EH		1XZ		1	1		1	A	T	I	YB
23				XX								E	XX		YC
24				YY								N	N		YC

ABSTRACT

DURING TESTING ON JUNE 24, JULY 13, AND JULY 15, 1981, 2B DIESEL GENERATOR FAILED TO MEET THE STARTING TIME REQUIREMENTS OF TECH SPEC 3.8.1.1. ON EACH OCCASION, THE DIESEL WAS DECLARED INOPERABLE AND ACTION STATEMENT 3.8.1.1 WAS ENTERED. SEE: 81-02. THE DIESEL FAILURE WAS DUE TO PROBLEMS WITH THE TURBO BOOST SYSTEM SOLENOIDS AND WAS REPAIRED. A SAFETY EVALUATION WAS PERFORMED DUE TO REPEATED DIESEL FAILURES. THE SAFETY EVALUATION FOUND THAT PROBLEMS WITH THE TURBO BOOST SYSTEM WERE THE ONLY COMMON SOURCE OF REPETITIVE DIESEL FAILURE. PREVENTIVE MAINTENANCE RECOMMENDATIONS GENERATED BY THE SAFETY EVALUATION HAVE BEEN IMPLEMENTED, AND 2B DIESEL HAS BEEN SATISFACTORILY TESTED.

Option 3: Producing a Barchart

Function The barchart provided by this option will give you a rough idea of how the steps occur as a function of event date; i.e., the bar chart presents the count of steps satisfying the searching logic as a function of the event date.

Syntax

```
STEPSCAN * BARCHART
```

The option is selected by including the term **BARCHART** in the STEPSCAN command line. By default, the scale on the date axis is by quarter of the year, although a monthly date bin can be selected by including the term **MONTH** in the command line.

```
STEPSCAN * MONTH
```

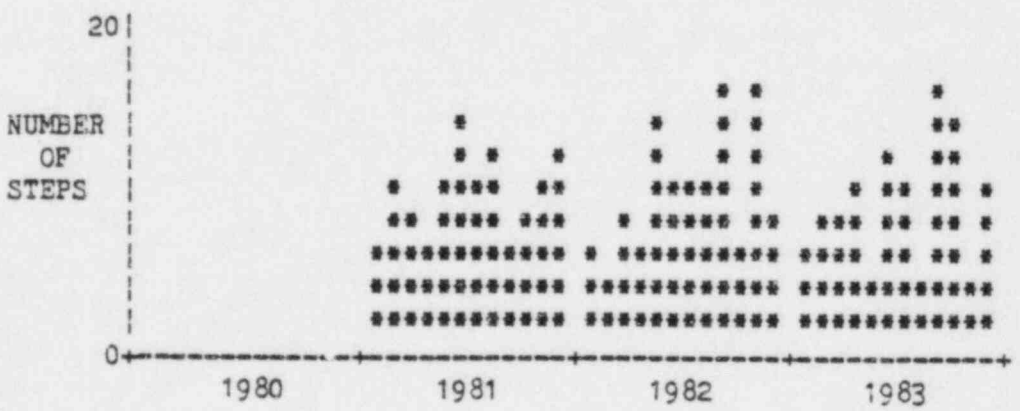
Note that the term **BARCHART** is not required when the term **MONTH** is used.

Example

```
*
STEPSCAN * MONTH
==>GROUP  4 HAS   328 LERs OUT OF   328 SEARCHED--OPERATION: "STEPSCAN"
    351 STEPS SATISFIED THE SEARCH STRATEGY
              SECONDS:      6.04 (CPU)     89.41 (CLOCK)--RATIO:0.068

TREND CHART

MAXIMUM COUNT FOR ANY MONTH  :    16
MINIMUM COUNT FOR ANY MONTH  :     0
          VERTICAL INCREMENT:     2
```



Option 4: Capturing Steps

Function An option is provided in the STEPSCAN command to collect the steps that satisfy the searching logic into a scratch file that can be processed by the STEPLIST, STEPSORT, SVALUES, and SRELATE commands. There are also options (**BEFORE** and **AFTER**) for capturing steps that occur before or after the steps that satisfy the searching logic and that link to these steps. (The before and after options are also used in the STEPLIST, STEPSORT, SVALUES, and SRELATE commands to designate exactly which step collection is to be processed.)

By default, the option to capture steps that satisfy the searching logic is turned on. To turn the option off, include the term **NOCAPT** or **NOCAPTURE** on the same line as the STEPSCAN command. At present, the capture files are set to allow capturing up to 1000 steps, after which the option is turned off and other STEPSCAN processing continues. A message will be displayed when this occurs. Please note that the only effect of filling a capture file is to not capture steps. All other processing proceeds in its normal fashion. Subsequent STEPSCAN command executions will not be affected in any way.

Using the capture options may be desirable or necessary for several reasons.

1. Processing from a captured file is efficient, because all steps, by definition, meet the searching logic. Subsequent processing can be made with no screening.
2. Steps in a captured file can be sorted and listed arbitrarily on any or all fields.
3. Coarse component codes (ICOMP) can be assigned to all components and analyzed by count or relationship to other fields.
4. Step and sequence types can be assigned. This produces three new fields (ISEQ1, ISEQ2, ISEQ3--the sequence number, the sequence type, and the step type, respectively), which may then be treated as new fields in the coded step matrix.
5. The BEFORE and AFTER step collections offer the ability to analyze precursors or effects of a particular kind of step. For example, if you locate steps involving diesel engines (dsl), the BEFORE step collection contains all those steps occurring earlier in the sequence that have a direct linkage to the diesel step; thus, analyzing these steps is an analysis of the "causes" of the diesel steps. Conversely, the AFTER steps reveal the effects or results of the diesel steps. (Note that the linkages are not just on adjacent steps or on steps that

immediately precede a step; i.e., if step 1 links to step 2, which links to step 5, then step 1 is linked to step 5, as is step 2.)

Syntax **STEPSCAN collection**

Example This example shows the effects of capturing the BEFORE and AFTER step collections.

```
*
FIND
+
<comp> plo pno
+
END
==>GROUP  1 HAS      10 LERs--OPERATION: "FIND"
*
STEPSCAN BEFORE
+
<comp> plo <eff> tb
+
END
THE FOLLOWING "STEP" DATA COLLECTIONS WILL BE CAPTURED:
    THE STEPS SATISFYING THE CURRENT SEARCHING LOGIC
    STEPS OCCURRING BEFORE THE STEPS SATISFYING THE CURRENT
         SEARCHING LOGIC WHICH LINK TO THESE STEPS
==>GROUP 2 HAS 4 LERs OUT OF 10 SEARCHED--OPERATION: "STEPSCAN"
      6 STEPS SATISFIED THE SEARCH STRATEGY
              SECONDS:   0.26 (CPU)   2.80 (CLOCK)--RATIO:0.093
*
RECALL 1
==>LER GROUP  3 CONTAINS THE      10 LERs FROM GROUP  1
*
STEPSCAN AFTER
+
<comp> plo <eff> tb
+
END
THE FOLLOWING "STEP" DATA COLLECTIONS WILL BE CAPTURED:
    THE STEPS SATISFYING THE CURRENT SEARCHING LOGIC
    STEPS OCCURRING AFTER THE STEPS SATISFYING THE CURRENT
         SEARCHING LOGIC WHICH LINK TO THESE STEPS
==>GROUP 4 HAS 4 LERs OUT OF 10 SEARCHED--OPERATION: "STEPSCAN"
      6 STEPS SATISFIED THE SEARCH STRATEGY
              SECONDS:   0.26 (CPU)   3.07 (CLOCK)--RATIO:0.085
```

INVENTORY DATABASE COMMANDS

Function The SCSS contains two major databases:

1. the Production database, consisting of the coded step matrix, abstracts, etc., and
2. an Inventory database, which is used to record the status of LERs received at the NOAC.

Example Figure 4-2, a listing of the inventory record for a particular LER, will be used to describe the fields included in the inventory data records.

This particular LER has had three additional revisions following its initial submission (revision 0). Its current revision level is 3. The REPORT STATUS and SCS CODE entries indicate whether or not this LER should be coded. (The blank entries on this form indicate the LER is codeable.) The DCS NUMBER is the NRC Document Control System number assigned to each document reported to the NRC; NSICNO is an identifier assigned to the document by the NSIC; PAGES is the number of pages submitted; STEPS is the number of steps coded, based on the submission; CODER contains the initials of the person who coded the matrix; and TIME is the time in minutes it took to code the matrix. Other columns, which are no longer used, are ERRORS, SIG1, SIG2, and SIG3.

The dates contain a detailed record of where the LER is within the normal processing scheme; LETTER contains the letter date of the revision; RECEIPT is the date the revision was received at the NOAC; SEQ CODE is the date the LER was sequence coded; ADD QA is the date the LER was processed through a special QA computer program; ADD PROD is the date the revision was initially added to production; RV PROD contains the date of the last change to the production record for the particular revision; and RV INV contains the last date on which the information for the particular revision was modified in the inventory database itself.

The above definitions are current after March 15, 1984. Inventory records entered before that time were made on another database management system, and the dates had slightly different meanings, especially in the latter two columns.

FIGURE 4-2. EXAMPLE INVENTORY RECORD

SCSS INVENTORY FORM 260/82-028 05-03-84

DOCKET 260 YEAR 82 LER NUMBER 028 CURRENT REVISION NUMBER: 3
REPORT STATUS: EVENT DATE: 9/21/1982
SCS CODE:

REV	DCS NUMBER	NSICNO	NUMBER OF PAGES	STEPS	ERRORS	CODER	TIME	SIG1	SIG2	SIG3
0	8210260446	0	3	12	0	RHG	15	0	0	0
1	8304190489	0	3	12	0	ABC	10	0	0	0
2	8311010099	0	3	10	0	DKL	25	0	0	0
3	8403200171	0	3	0	0		0	0	0	0

PROCESSING DATES FOR THIS LER

REV	LETTER	RECEIPT	SEQ CODE	ADD QA	ADD PROD	RV PROD	RV INV
0	10/19/82	11/11/82	12/ 9/82	1/14/83	1/21/83	9/20/83	0/ 0/ 0
1	4/14/83	5/ 9/83	8/ 8/83	9/ 9/83	9/20/83	2/ 2/84	0/ 0/ 0
2	10/19/83	11/30/83	1/ 4/84	1/26/84	2/ 2/84	2/ 2/84	2/ 2/84
3	3/ 9/84	4/ 4/84	0/ 0/ 0	0/ 0/ 0	0/ 0/ 0	0/ 0/ 0	4/ 4/84

INVENT

Function The inventory record shown in Fig. 4-2 was listed by the INVENT command, which serves for this and several other purposes.

NOTE: Because the INVENT command can require several minutes (unless you are just listing an inventory record), your screen will display messages concerning the number of LERs processed so that you will know that the command is operating.

Syntax

INVENT LIST ddd yy nnn

INVENT is a one-line command. To list an inventory record for an LER, follow the term INVENT by the term LIST, then ddd yy nnn, where ddd is the docket number, yy is the year, and nnn is the LER number.

By specifying a date range, you can use INVENT to produce a summary report concerning LER processing within that date range.

INVENT RANGE mm/dd/yy mm/dd/yy

Follow the term INVENT by **RANGE** then the beginning and ending dates for the period, where the dates are in the form mm/dd/yy (mm is the month, dd is the day, and yy is the year).

INVENT DOCKET=ddd YEAR=yy

You can restrict the INVENT command to LERs for a particular docket or year by including the terms **DOCKET=ddd** and **YEAR=yy** in the command line, where ddd contains the docket number and yy contains the year. This works only with YEAR=xx when a range is specified.

An additional capability of the INVENT command is to be able to capture the record keys for LERs with a particular kind of date (from those mentioned above) that falls within the date range specified for the command. You can enter INVENT followed by CAPTURE followed by one of the terms shown in the following table.

Date Term	LER has a date in the period on at least one revision for
LETTER	Letter date
EVENT	Event date
SEQ	Sequence coding date
RECD	Receipt date at NOAC
QA	QA processing date
PROD	Date added to the production file
RVPROD	Date the production file was changed
RVINV	Date an inventory record was changed

Note that the CAPTURE option does not affect other parts of the command.

Example 1 This report is produced very quickly, because the command can restrict itself to processing only subsets of the total inventory database.

```
•
INVENT RANGE 1/1/83  12/31/83  DOCKET=029 YEAR=83

INVENTORY REPORT FOR DOCKET:029 YEAR:83

            REPORT PERIOD:  19830101 TO  19831231
-------------------------------------------------
       NUMBER OF EVENTS OCCURRING IN PERIOD:   42
NUMBER OF LETTER DATES WHICH FALL IN PERIOD:   43
                 NUMBER OF LERS RECEIVED:      41
               NUMBER RECEIVED FOR CODING:     41
             NUMBER OF LERS SEQUENCE-CODED:    35
              NUMBER ADDED TO THE QA FILE:     33
  NUMBER OF CHANGES TO THE PRODUCTION FILE:    25
    NUMBER OF NEW LERS ADDED TO PRODUCTION:    24
              NUMBER OF OLD LERS CHANGED:       0
  NUMBER OF CHANGES TO THE INVENTORY FILE:      2
             NUMBER OF REPLACED REVISIONS:      1
NUMBER OF LERS IN FILE AT PERIOD BEGINNING:     0
   NUMBER OF LERS IN FILE AT PERIOD ENDING:    24
```

Example 2 This example shows the record keys for LERs added to the production database in March 1984 for docket 029.

```
*
INVENT RANGE 3/1/84  3/31/84  DOCKET=029 CAPTURE=prod

INVENTORY REPORT FOR DOCKET:029 YEAR:ALL

            REPORT PERIOD:  19840301 TO  19840331
-------------------------------------------------
       NUMBER OF EVENTS OCCURRING IN PERIOD:    0
NUMBER OF LETTER DATES WHICH FALL IN PERIOD:    1
                  NUMBER OF LERS RECEIVED:      1
               NUMBER RECEIVED FOR CODING:      1
             NUMBER OF LERS SEQUENCE-CODED:     0
              NUMBER ADDED TO THE QA FILE:      2
  NUMBER OF CHANGES TO THE PRODUCTION FILE:     2
    NUMBER OF NEW LERS ADDED TO PRODUCTION:     2
               NUMBER OF OLD LERS CHANGED:      0
   NUMBER OF CHANGES TO THE INVENTORY FILE:     3
             NUMBER OF REPLACED REVISIONS:      0
NUMBER OF LERS IN FILE AT PERIOD BEGINNING:   110
   NUMBER OF LERS IN FILE AT PERIOD ENDING:   112

   2 CLASS:PROD     LERS ARE IN THE "ACTIVE" RECORD GROUP

*
DISPLAY
  1  029/83-038   2  029/83-039
```

SINVENT

Function The SINVENT command looks at all entries in the database and produces a table that summarizes the overall status of the system regarding total contents of the database and the number of LERs that need additional processing before they are included in the production database.

Syntax

SINVENT SAVE or **OLD**

SINVENT is a one-line command, consisting of the term SINVENT and, optionally, either **SAVE** or **OLD.** When the term SINVENT is used by itself, the program scans the database and writes out the prescribed table, after which it returns to the command entry level. When the term **SAVE** is included, the results of using the command are recorded in your work area. Subsequent executions of the command will automatically pick up saved entries and include these lines in the new table, thereby providing a record on how the overall status of the database varies with time. Using the term **OLD** as a second term in the SINVENT command line causes the program to skip new processing and list only entries saved on previous executions.

NOTE: SINVENT requires several minutes for processing, except for the special SINVENT OLD command form.

Example

```
*
SINVENT

**********************************************************************
THIS COMMAND SCANS THE ENTIRE DATABASE AND REQUIRES 5-15 MINUTES
**********************************************************************

          *================================*======*
          * LERS TO BE CODED, KEYPUNCHED,  * LERS *
*=========*       OR ADDED TO QA FILE      *  IN  *=======*=======*=====*
* REPORT *====*====*======*======*=======*  QA  * STEPS *  LERS * NO  *
*  DATE  * 81 * 82 *  83  *  84  * TOTAL * FILE * IN DB * IN DB * ABS *
*=========*====*====*======*======*=======*======*=======*=======*=====*
 05-03-84  21   20    178    307     526    199   76936   12934     0
```

STEPSORT

Function The STEPSORT command allows sorting the data in a captured data file based on any or all of the fields listed in the preceding table.

Syntax

STEPSORT collection <field1> <field2>

STEPSORT is a one-line command consisting of the term STEPSORT, followed by the name of the collection to be sorted, followed by the names of the fields to be sorted. Omission of the collection name (*, BEFORE, or AFTER) causes the command to process the * collection (i.e., those steps that match the STEPSCAN searching logic).

If all fields to be sorted cannot be accommodated in a single line, you can enter multiple lines.

Example

In this example, the collection is the * collection. This sort was performed according to the hierarchy: field1, field2,.....field20.

```
*
STEPSORT  field16  field17  field18  field19  field20
.
.
.
.
*
STEPSORT  field11  field12  field13  field14      field15
.
.
.
.
*
STEPSORT     field6   field7   field8   field9    field10
.
.
.
.
*
STEPSORT     field1   field2   field3   field4    field5
```

The STEPSORT command will be further illustrated in the examples for the STEPLIST command.

STEPLIST

Function To obtain a simple listing of the data in one of the captured data collections, you can use the STEPLIST command.

Syntax

STEPLIST collection

Type STEPLIST followed by the name of the collection to be listed (*, BEFORE, or AFTER). Omission of the collection name selects the * collection.

Example 1 In the case shown in Fig. 4-3, the options in the STEPSCAN step were set to capture all three step collections--*, BEFORE, and AFTER--and the ICOMP, ISEQ1, ISEQ2, and ISEQ3 fields were assigned. The data for the * collection were then listed. In the * collection, the ICOMP values were sorted.

Example 2 The example given in Fig. 4-4 shows the sort of the AFTER collection (those steps that followed the pipe hanger steps) according to the ICOMP field, after which a STEPLIST command shows the effects of the sort.

FIGURE 4-3. EXAMPLE OF STEPSCAN AND STEPLIST

```
*
FIND
+
<comp> hang
+
END
==>GROUP  1 HAS  39 LERs--OPERATION:  "FIND"
*
STEPSCAN * <icomp> <iseq> BEFORE AFTER
==>GROUP 2 HAS 39 LERs OUT OF 39 SEARCHED--OPERATION: "STEPSCAN"
    41 STEPS SATISFIED THE SEARCH STRATEGY
          SECONDS:  2.42 (CPU)  45.60 (CLOCK)--RATIO:0.053
*
STEPLIST

THE FOLLOWING TABLE IS FOR STEPS WHICH MATCH THE
CURRENT SEARCHING LOGIC

DOC YR LER R ST LK SL C  S  IS COMP VEND QUAN TR CH DI TP D E  IC  SEQ-ID
--- -- --- - -- -- -- -- -- -- ---- ---- ---- -- -- -- ---- -- --- ------
237 81 018 0 2  1     RC BI     HANG      M    1     1  MTRI AX 310  1 1 3
237 81 057 0 3  1  A  RC BN     HANG I207 2    1     1  MTRI AX 310  1 7 3
237 81 061 1 4  3     CA CB     HANG I206 1    1     1  MT H DA 310  1 1 7
265 81 025 0 3  A     XA HD     HANG G255 2    1     1  PT H CA 310  1 7 4
289 81 007 0 3  1  A  RC ZY     HANG G150 M    M     1  MP L AE 310  1 6 3
312 81 029 0 1  0     ZZ FA     HANG B209 3    3     1  MT I AB 310  1 3 1
317 81 029 0 2  1     CA CB     HANG B130 Z    Z     1  PT H DA 310  1 2 7
325 81 034 0 4  1  A  RC ZY     HANG      2    M     1  MTRL KX 310  1 6
327 81 069 0 2  1     RC BK     HANG      1    1     1  MTRL AA 310  1 9 3
327 81 069 0 7  A     RC BK     HANG      1    2     2  MTRL AA 310  1 9 4
327 81 072 0 2  1     RC CB     HANG      1    1     1  ATRL AA 310  1 1 7
334 81 001 0 5  A     RC FA     HANG      3    1     1  PT F CA 310  1 9 4
336 81 008 0 2  1     RC AD     HANG      1    1     1  MTRR AE 310  1 1 3
336 81 032 0 3  A     RC ZY     HANG      M    Z     1  PT F DX 310  1 5 6
338 81 035 1 1  0  A  XA ZY     HANG G255 7    M     1  MT H CA 310  1 9 1
338 81 035 1 8  7  B  XA ZY     HANG      13   M     2  MT H CA 310  1 9 3
366 81 104 1 3  1     RC BN     HANG      1    1     1  MTRI DB 310  1 1 7
366 81 120 0 1  0  A  ZZ BN     HANG      1    1     1  AT Z DB 310  1 9 1
368 81 037 0 2  1     RC BF     HANG B130 1    1     1  MTRP AA 310  1 1 3
369 81 025 0 2  1     RC EN     HANG      12   Z     1  MTRR PL 310  1 2 7
250 82 007 0 1  0  A  FK WF     HANG      Z    1     1  MT I DA 310  1 7 1
302 82 044 0 2  1     RC EN ZZ  HANG      3    Z     1  MTRR XX 310  1 2 3
331 82 014 0 5  A  B  FK BH     HANG      1    1     1  AT I DA 310  1 7 4
331 82 022 0 3  2     RC BN     HANG B209 1    1     1  PP L KH 310  1 1 3
346 82 033 0 3  2  A  RC AE     HANG      1    1     1  PTRL KF 310  1 9 3
361 82 032 0 2  1     RC BA     HANG      2    1     1  ATRI DB 310  1 1 7
366 82 018 1 3  2     RC FA SE  HANG B209 1    1     1  ATRI AB 310  1 1 7
369 82 016 0 2  1  A  RC BK     HANG      1    1     1  MTRI KT 310  1 9 3
```

FIGURE 4-4. EXAMPLE OF STEPSORT AND STEPLIST

```
*
STEPSORT AFTER <icomp>
*
STEPLIST AFTER

THE FOLLOWING TABLE IS FOR LATER STEPS THAT LINK TO
STEPS WHICH MATCH THE CURRENT SEARCHING LOGIC

DOC YR LER R ST LK SL C  S  IS COMP VEND QUAN TR CH DI TP D E  IC  SEQ-ID
--- -- --- - -- -- -- -- -- -- ---- ---- ---- -- -- -- ---- -- --- -------
366 81 120 0 3  A        XX                              EXX  YB      1 9 8
302 82 044 0 3  2        XX                              EAA  YB      1 2 7
309 83 002 1 7  B        YY                              NK   YB      1 9 8
369 83 016 0 3  A        XX                              XXX  YB      1 8 8
395 82 032 0 5  A     HP WA SN TK   U081 1     1      1   AT F BP 10   1 4 4
369 83 023 0 2  1     SX PO    PUX       Z             A   MT R UB 270  1 1 3
237 81 018 0 3  2     CB BI SC PIAZ      1     1      1   MP I BK 280  1 1 3
237 81 057 0 6  A     RC BN ZZ PIEZ      1     1      1   PT F BP 280  1 7 4
265 81 025 0 4  3     CA HD ZZ PIZZ      1     1      1   PT H BP 280  1 7 3
289 81 007 0 5  A     RC ZY ZZ PIZZ      M     M      1   PT F BP 280  1 6 8
312 81 029 0 2  1     RC FA SL PIZZ      2     M      1   PT F BP 280  1 3 3
325 81 034 0 5  A     RC BX ZZ PIZZ      3     M      1   PT F BP 280  1 6 8
325 81 034 0 6  A     RC BH ZZ PIDZ      7     M      1   PT F BP 280  1 6 8
325 81 034 0 7  A     RC BN ZZ PIEZ      2     M      1   PT F BP 280  1 6 8
327 81 069 0 3  2     RC BK ZZ PIAZ      1     1      1   PT F BP 280  1 9 7
327 81 069 0 8  7     RC BK ZZ PIAZ      1     2      2   PT F BP 280  1 9 7
336 81 008 0 3  2     CA AD SA PIZZ      1     1      1   PT F BP 280  1 1 3
336 81 032 0 4  3     RC FS    PIZZ      1     Z      1   AT X KT 280  1 5 7
336 81 032 0 5  3     RC BL    PIZZ      1     Z      1   AT X KT 280  1 5 7
338 81 035 1 4  A     CA ZY ZZ PIZZ      M     M      1   PT F BP 280  1 9 8
338 81 035 1 9  B     CA ZY ZZ PIZZ      M     M      1   PT F BP 280  1 9 8
368 81 037 0 3  2     RC BF ZZ PIZZ      1     1      1   PT P BP 280  1 1 7
250 82 007 0 3  A     RC WF    PIZZ      1     1      1   MT I AE 280  1 7 8
346 82 033 0 5  A     RC AE SA PIDZ      1     1      1   PT F BP 280  1 9 8
369 82 016 0 4  A     RC BK SA PIZZ      Z     Z      1   PT F BP 280  1 9 8
324 83 090 0 4  A     RC WK SE PIAZ      2     M      1   PT F BP 280  1 9 8
324 83 090 0 5  A     RC KC SE PIAZ      1     1      1   PT F BP 280  1 9 8
327 83 162 0 4  3     CH KF SH PIZZ      M     M      1   PT F BP 280  1 9 3
334 81 001 0 6  5     RC FA    WALL      M     1      1   PT F DA 320  1 9 3
331 82 022 0 4  3     RT BN    XXX                     1   AT L KT 920  1 1 7
331 82 014 0 7  B     RS BH    1XZ       1     1      1   AT I KT 930  1 7 4
```

SVALUES

Function

The VALUES command (Chap. 3) is an extremely useful command for determining the usage of codes in a particular set of steps. Its counterpart for dealing with the captured step data collections is the SVALUES command.

One of the more interesting capabilities of the SVALUES command is the ability to obtain values on the BEFORE and AFTER data collections. The BEFORE collection represents precursor steps. Thus, analyzing this collection is equivalent to looking at earlier causes of a given event. Example 2 determines the codes in the component field that preceded, and contributed to, pipe hanger failures.

The probabilities presented in an analyses of the BEFORE and AFTER collections are useful to the probabilistic risk analyst because they provide "real" evaluations of branching fractions, etc., quantities that normally would be based on considerable conjecture.

Syntax

SVALUES collection <field1> <field2>

This command is a one-line command giving the term SVALUES followed by the name of the captured data collection--*, BEFORE, or AFTER--followed by the names of the fields to be analyzed and including any special processing option terms that you wish. Omission of the name of the captured data collection automatically selects the * collection.

The processing options for this command are:

1. KEYSORT-to cause the codes for the fields occurring after this term in the command line to be sorted alphabetically, and
2. VSORT-to cause the codes for the fields occurring after this term in the command line to be sorted by count in descending order.

By default, code listings are always sorted by count.

Options are provided to limit the size of the tables produced by SVALUES. Use of **TOP** n on the command line limits the output to the n codes with the highest counts for the fields following these terms, whereas **OVER n** limits the output to those codes with counts larger than n (see VALUES).

Example 1 This example determines all codes in the cause field in the * collection.

```
*
SVALUES <cause>

THE FOLLOWING TABLE IS FOR STEPS WHICH MATCH THE CURRENT
SEARCHING LOGIC

THE ACTIVE LIST OF 39 LERs HAS 8 UNIQUE VALUES IN THE CAUSE
FIELD FOR STEPS THAT MATCH THE CURRENT LOGIC TABLE

KEY VALUE   NUMBER OF STEPS        DESCRIPTION
    RC        23 (56.1%)    RESULTANT COMPONENT FAULT
    FK         7 (17.1%)    WATER HAMMER
    CA         3 ( 7.3%)    MECHANICAL OVERLOAD
    XA         3 ( 7.3%)    CHANGE IN REGULATORY REQUIREMENTS
    ZZ         2 ( 4.9%)    UNKNOWN
    AD         1 ( 2.4%)    WRONG MATERIAL
    CB         1 ( 2.4%)    FATIGUE
    AC         1 ( 2.4%)    WRONG PART
```

Example 2 To determine the codes in the component field of the steps preceeding the pipe hanger steps, the following command is given. (The original FIND command asked for pipe hangers in the component field; see Fig. 4-3.)

```
*
SVALUES BEFORE <comp>
```

THE FOLLOWING TABLE IS FOR EARLIER STEPS THAT LINK TO STEPS WHICH MATCH THE CURRENT SEARCHING LOGIC

THE ACTIVE LIST OF 39 LERs HAS 16 UNIQUE VALUES IN THE COMP FIELD FOR STEPS THAT MATCH THE CURRENT LOGIC TABLE

KEY VALUE	NUMBER OF STEPS	DESCRIPTION
PZ	10 (23.8%)	UNKNOWN PERSONNEL
FAS	5 (11.9%)	FASTENER
PUX	5 (11.9%)	OTHER/UNKNOWN UTILITY PERSONNEL
PIZZ	4 (9.5%)	PIPE, UNKNOWN DIAMETER, UNKNOWN MATERIAL
XXX	4 (9.5%)	ENTIRE SYSTEM
MSC	3 (7.1%)	MISCELLANEOUS SUBCOMPONENT
PCP	2 (4.8%)	CONTRACTOR PERSONNEL
328	1 (2.4%)	
MEI	1 (2.4%)	MISCELLANEOUS EQUIPMENT ITEM
339	1 (2.4%)	
SHFT	1 (2.4%)	SHAFT/STEM
PMPZ	1 (2.4%)	PUMP, UNKOWN TYPE
CSTR	1 (2.4%)	CONCRETE STRUCTURE/SHIELD
DFM	1 (2.4%)	DIAPHRAM
FCVZ	1 (2.4%)	VALVE, CONTROL, FLOW, UNKNOWN MATERIAL
PIEZ	1 (2.4%)	PIPE, 16 INCH AND LARGER DIAMETER, UNKNOWN MATERIAL

From this, you would immediately suspect personnel actions as being a strong contributor to pipe hanger failures.

SRELATE

Function The RELATE command (Chap. 3) produces a two-dimensional matrix showing how pairs of codes (any two fields) are used in particular steps. Its counterpart for the captured step data collections is the SRELATE command.

Syntax

SRELATE collection <field1> <field2>

SRELATE is a one-line command consisting of the term SRELATE followed by the name of the collection to be processed--*, BEFORE, or AFTER--followed by the names of the two fields to be processed, along with other terms that are used to select special processing options for the command. Omission of the name of the captured data collection automatically selects the * collection.

Two special processing options exist for this command.

1. KEYSORT-to cause the two-dimensional tables to be sorted alphabetically by code (by default, the tables are sorted by counts). Both fields are always sorted in the same way.
2. PERCENT-to request the units inside the table to be in percentages, as opposed to step count.

Options are provided to limit the output from SRELATE. These are the **TOP** and **OVER** options used in many other SCSS commands (see RELATE). Including **TOP n** in the SRELATE command line limits the table to those n codes with the highest counts, whereas **OVER n** limits the table to those codes with counts greater than n. Because there are two fields to deal with, the use of these options is more involved than for many commands. The <u>first</u> use of these options applies to the <u>first</u> field on the command line; the second use applies to the second field.

SRELATE <cause> <eff> TOP5 TOP3

This would limit the table to the five top cause codes and the three top effect codes.

Example From the analysis shown in the Fig. 4-5, you can see that unknown procedural problems (CAUSE=SZ) involving unknown personnel were the most prevalent precursor to pipe hanger failures.

FIGURE 4-5. EXAMPLE OF SRELATE

```
*
SRELATE BEFORE <comp> <cause>
FIELD:COMP     HAS  16 UNIQUE VALUES
FIELD:CAUSE    HAS  18 UNIQUE VALUES

THE FOLLOWING TABLE IS FOR EARLIER STEPS THAT LINK TO
STEPS WHICH MATCH THE CURRENT SEARCHING LOGIC
```

*COMP * * * * CAUSE*	PZ	FAS	PUX	PIZZ	XXX	MSC	PCP	328	MEI
SZ	6	0	2	0	0	0	1	0	0
RC	0	2	0	2	0	3	0	0	1
PW	0	0	0	0	4	0	0	0	0
SX	2	0	1	0	0	0	0	0	0
FK	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0
SH	0	0	2	0	0	0	0	0	0
ZZ	0	1	0	0	0	0	0	0	0
FE	0	1	0	0	0	0	0	0	0
SK	1	0	0	0	0	0	0	0	0
PY	0	0	0	1	0	0	0	0	0
XX	0	0	0	1	0	0	0	0	0
AD	0	0	0	0	0	0	0	0	0
SB	0	0	0	0	0	0	1	0	0
DB	0	0	0	0	0	0	0	0	0
HK	0	0	0	0	0	0	0	0	0
SA	1	0	0	0	0	0	0	0	0
AC	0	0	0	0	0	0	0	0	0
SUM	10	5	5	4	4	3	2	1	1

FIGURE 4-5. Continued

THE FOLLOWING TABLE IS FOR EARLIER STEPS THAT LINK TO
STEPS WHICH MATCH THE CURRENT SEARCHING LOGIC

*COMP * * * * CAUSE*	339	SHFT	PMPZ	CSTR	DFM	FCVZ	PIEZ	
SZ	0	0	0	0	0	0	0	9
RC	0	0	0	0	0	0	0	8
PW	0	0	0	0	0	0	0	4
SX	0	0	0	0	0	0	0	3
FK	0	1	0	0	0	0	0	2
	1	0	0	0	0	0	0	2
SH	0	0	0	0	0	0	0	2
ZZ	0	0	1	0	0	0	0	2
FE	0	0	0	0	0	0	0	1
SK	0	0	0	0	0	0	0	1
PY	0	0	0	0	0	0	0	1
XX	0	0	0	0	0	0	0	1
AD	0	0	0	1	0	0	0	1
SB	0	0	0	0	0	0	0	1
DB	0	0	0	0	1	0	0	1
HK	0	0	0	0	0	1	0	1
SA	0	0	0	0	0	0	0	1
AC	0	0	0	0	0	0	1	1
SUM	1	1	1	1	1	1	1	42

JOIN

Function The sublink field is used whenever two or more steps combine to cause a step. The JOIN command provides a means of making searches based on this field.

Syntax

JOIN

At each + prompt, insert the case you are interested in.

case 1
+
case 2
+
case 3

JOIN uses options 2, 3, and 4, discussed on pp. 3-14 through 3-21, to specify its searching logic. For the JOIN command, the syntax must always describe at least three cases, which will be interpreted as follows:

> find LERs that have a step that satisfies case 1 and a step that satisfies case 2 and that have other steps that satisfy all cases before the last case, which are linked through a single sublink value to a step that satisfies the last case (e.g., for three cases, the logic is interpreted as find those LERs that have case 1 and case 2 combining to cause case 3).

JOIN RANGE date1 date2

The JOIN command can be restricted to processing LERs in a specific date range by including the terms **RANGE date1 date2** in the JOIN command line. The dates are specified as mm/dd/yy, where mm is the month, dd is the day, and yy is the year.

NOTE: The JOIN command, although similar to the LINK command, requires direct linkages to produce a hit. A case that links to an intermediate case that links to the final case will not count. All cases must link to the final case.

Example Figure 4-6 is an example of the JOIN command.

FIGURE 4-6. EXAMPLE OF JOIN

```
*
FIND
+
<comp> hang spt .beg. pi
+
END
*
JOIN
+
<comp> hang
+
<comp> spt
+
<comp> .beg. pi
+
END
==>GROUP 2 HAS 3 LERS OUT OF 3 SEARCHED--OPERATION: "JOIN"
            SECONDS: 0.25 (CPU)   5.07 (CLOCK)--RATIO:0.049
*
OPTIONS LIST MATRIX NOABST
*
LIST 1
FORM     1                  LER SCSS DATA                          05-03-84
*****************************************************************************
DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER    NSIC      EVENT DATE
  237   1981       057          0  8109230144  168894       9- 3-1981


COMMENTS
 POSSIBLE DESIGN ERROR - NO DRAIN LINE FOR THE HPCI PIPE THUS WATER WAS
 BUILDING UP DURING LINE WARMUP.

DOCKET:237  DRESDEN 2                    TYPE:BWR
                REGION:  3             VENDOR:GE
ARCHITECTURAL ENGINEER: S&L
     FACILITY OPERATOR: COMMONWEALTH EDISON CO.
                SYMBOL: DRS

REFERENCE LERS:
    1 237/70-035

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0       FE   BN       FAS         M    1     1 M T  K  AB
  2   1   X   FK   BN       FAS         M    1     1 M T  K  AB
  3   1   A   RC   BN       HANG  I207  2    1     1 M TR I  AX
  4   0   A   FE   BN       SPT         2    1     1 M T  I  DA
  5   4   X   FK   BN       SPT         2    1     1 M T  I  DA
  6   A       RC   BN   ZZ  PIEZ        1    1     1 P T  F  BP
  7   6       RT   BN       XXX                     1 A T  I  KT
  8            XX                                     E XX    YC
  9            YY                                     N N     YC
```

LIST, FLOW CHART OPTION

Function A special option is provided in the LIST command to produce a flow chart that shows how the steps in the coded step matrix are related to each other. These charts allow you to easily recognize events involving two or more independent sequences, potential common-cause steps, etc.

Syntax

OPTIONS LIST FLOW

To select the option, the OPTIONS command should include the term **FLOW**, after which any LIST command will produce the flow chart until the option is cancelled by using **NOFLOW**.

Example 1 This is a typical "cascading" type event, wherein each step was a result of the one immediately preceding it. Note that initiating step (step 1) will use equal marks (=) on the horizontal lines. In the vertical lines, the step number at the top of the lines "causes" all steps below it, except when two or more steps join to cause one or more other steps, in which case the same letter (sublink) will occur two or more times to the right of the step numbers that combine.

```
*
OPTIONS LIST FLOW NOCOMM NOWATCH NOREF NODOCK
*
LIST 1

FORM     1                  LER SCSS DATA                          05-03-84
*****************************************************************************
DOCKET   YEAR   LER NUMBER   REVISION   DCS NUMBER     NSIC       EVENT DATE
  029    1981       012          0     8108210332    168282       6- 2-1981

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0       SC   PM        PX         Z          A A T  F  UA
  2   1       RC   EA        TLIN       1    1     1 A T  F  KU
  3   2       RC   IN   KB   MOT        2    Z  M  1 A T  F  KH
  4   3       RC   IN   KB   PMPZ       2    Z  M  1 A T  F  KH
  5   4       RC   IN   KB   RI         2    Z  M  1 A T  F  LE
  6            XX                                  H XX     YC
  7            YY                                  N N      YC

STEP INTERCONNECTION DIAGRAM

 1==>1
     ¦
     ¦
     2 -->2
          ¦
          ¦
          3 -->3
               ¦
               ¦
               4 -->4
                    ¦
                    ¦
                    5
ABSTRACT
 WHILE PERFORMING ROUTINE SURVEILLANCE, A CHEMISTRY DEPARTMENT
 TECHNICAL ASSISTANT DISCOVERED THAT NUMBERS 1 AND 7 CONTINUOUS
 ENVIRONMENTAL AIR SAMPLER PUMPS WERE NOT RUNNING.  TECH. SPEC.
 REQUIRES THAT 6 OUT OF THE 7 MONITORS BE OPERABLE AT ALL TIMES.  THIS
 IS THE FIRST OCCURRENCE OF THIS NATURE.  THE CAUSE WAS LOSS OF POWER
 TO THE MONITOR PUMP DRIVE MOTOR RESULTING WHEN THE LOCAL UTILITY
 PERFORMED MAINTENANCE ON THE FEEDER LINE.  UPON COMPLETION OF THE
 MAINTENANCE THE LINE WAS ENERGIZED AND THE MONITORS WERE OPERABLE. THE
 LOCAL UTILITY HAS AGREED TO NOTIFY THE PLANT PRIOR TO FUTURE OUTAGES
 SO AN ALTERNATE POWER SUPPLY CAN BE USED.
```

Example 2 This diagram shows an event with two separate and independent sequences.

```
•
LIST 102

FORM   102                    LER SCSS DATA                          05-03-84
*****************************************************************************
DOCKET  YEAR  LER NUMBER  REVISION  DCS NUMBER    NSIC      EVENT DATE
  366   1981      094         0     8110290168   169931      9-21-1981

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0       ZZ   IH   EH  SEAL       1    1  1  1 M T  K  BP
  2   1       RC   IH   EH  65         1    1  1  1 A TR I  LF
  3   2       RC   EH      DSL  F010   1    1     1 A T  I  KN
  4   3       RS   EH      1X3         1    1     1 A T  I  YB
  5           XX                                     I XX    YC
  6           YY                                     N N     YC
  7   0       ZZ   IH   EH  MSC        1    1  1  1 A T  K  L
  8   7       RC   IH   EH  MOT        1    1  1  1 A TR L  KF
  9   8       RC   IH   EH  65         1    1  1  1 A TR F  LE
 10   9       RC   EH      DSL  F010   1    1     1 A T  F  KI
 11   10      RS   EH      1X3         1    1     1 A T  F  YB
 12           XX                                     E XX    YC
 13           YY                                     N N     YC

STEP INTERCONNECTION DIAGRAM

 1==>1
     ¦
     ¦
     2 -->2
          ¦
          ¦
          3 -->3
               ¦
               ¦
               4



 7==>7
     ¦
     ¦
     8 -->8
          ¦
          ¦
          9 -->9
               ¦
               ¦
               10-->10
                    ¦
                    ¦
                    11
```

Example 3 In this example, there are three situations (denoted by sublink values A, B, and C) which illustrate two steps combining together to cause a third step. The group of steps 6, 7, 17, and 21 illustrates a potential common-cause situation where step 6 or 7 could have been the common-cause step.

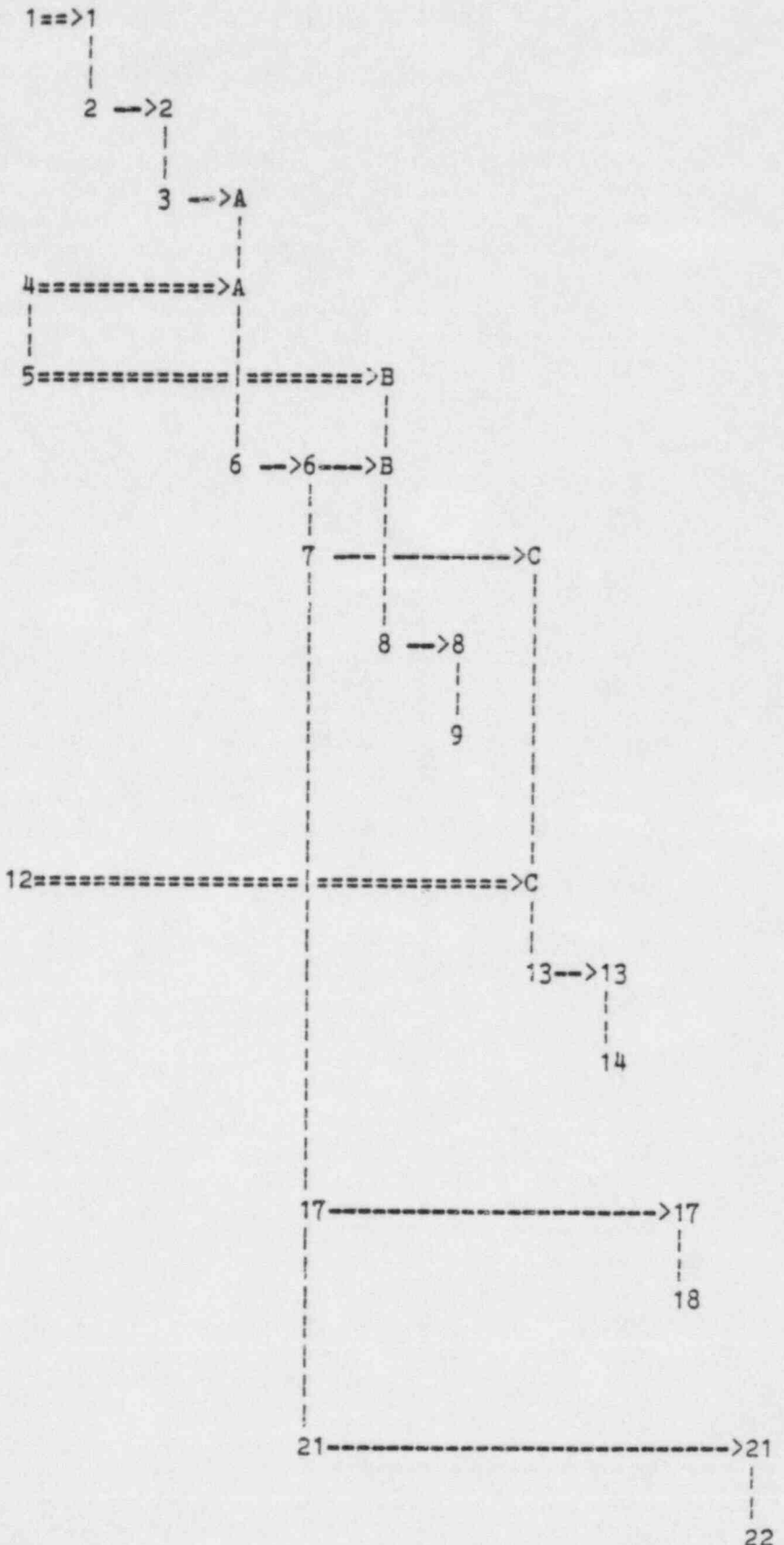
```
*
list 58

FORM     58                    LER SCSS DATA                          05-03-84
**********************************************************************************
DOCKET   YEAR   LER NUMBER   REVISION   DCS NUMBER    NSIC      EVENT DATE
  311    1981       050          1     8207080364   175161      6-25-1981

STEP LK SLK CAUSE PSYS ISYS COMP VEND QUAN TR CH DI T P  D EFF
  1   0       SZ   PI       PZ          Z          A M T  K  UB
  2   1       RC   CK       CBL         M    1     1 M TR K  AE
  3   2   A   RC   CK       SOL  A499   1    1     1 M TR K  KF
  4   0   A   ZZ   CK       SOL  A499   1    1     2 M T  K  KF
  5   0   B   ZZ   CK       MOT         1    1     1 A T  I  KF
  6   A   B   RC   EH       TCHG        1    1     1 A T  I  KF
  7   6   C                                                  YC
  8   B       RC   EH       DSL         1    1     1 A T  I  KF
  9   8       RS   EH       1X2         1    1     1 A T  I  YB
 10               XX                                  L XX    YC
 11               YY                                  N N     YC
 12   0   C   ZZ   CI       MEI         1    1     1 A T  K  PE
 13   C       RC   EH       DSL         1    1     1 A TR I  KF
 14  13       RS   EH       1XZ         1    1     1 A T  I  YB
 15               XX                                  E XX    YC
 16               YY                                  N N     YC
 17   6       RC   EH       DSL         1    1     1 A TR I  KF
 18  17       RS   EH       1XZ         1    1     1 A T  I  YB
 19               XX                                  C XX    YC
 20               YY                                  N N     YC
 21   6       RC   EH       DSL         1    1     1 A TR I  KF
 22  21       RS   EH       1XZ         1    1     1 A T  I  YB
 23               XX                                  E XX    YC
 24               YY                                  N N     YC
```

Example 3 (Continued)

STEP INTERCONNECTION DIAGRAM



DEFINE AND DEFAULT

Function

As a user of the SCSS, you have been assigned a modest amount of work space within the database, within which you can save searching strategies and database status summaries.

Whenever you request a record, the system looks (by default) first at this work space, and, if it does not find it there, proceeds to look in the production files. This hierarchy of searching is extremely powerful, as it allows you to tailor the database for your personal situation without affecting any other user in the system. At present, most users do not need such tailoring, although it is very important to those who maintain the database because it allows them to add new records, test update operations, etc., before actually moving the new information to the production files.

Your work area has an assigned name: iiiA0000, where iii are the initials you used during the log on procedures. The production files are collectively referred to under the name STD, for Standard.

The DEFINE command is provided to allow resetting the default searching hierarchy. The DEFAULT command returns you to the standard searching hierarchy. The DEFINE and DEFAULT commands are not expected to be of much use to most users; however, there are situations where they will prove useful. Assume, for example, that user aaa knows that user bbb has constructed a searching strategy that user aaa wants to copy into his own work space under the same or a different name. User aaa could log on to the system and issue:

DEFINE bbbA0000

Then he would activate the searching strategy of interest and issue a SAVE command to store it in his own work space (a user only has write privileges to his own space).

Syntax

DEFINE work area

DEFINE is a one-line command, consisting of the term DEFINE, followed by up to four names of work areas or collections (e.g., STD) to be looked in.

DEFAULT

DEFAULT is a one-line command consisting only of the term DEFAULT.

Example In this example, which follows the above procedure, note that the DEFAULT command is issued after the searching sequence is activated to return the system to its normal searching hierarchy.

```
•
DEFINE xyza0000
THE FOLLOWING DATASETS WILL BE IN THE SEARCH HEIRARCHY
    XYZA0000
•
ACTIVE pwrdiese
LER GROUP PWRDIESE IS NOW "ACTIVE" WITH 190 LERs
•
DEFAULT
THE SEARCH HEIRARCHY WILL BE RESET TO ITS DEFAULT
•
SAVE pwrdiese
AN ACTIVE GROUP NAMED PWRDIESE WITH 190 LERS HAS BEEN SAVED
THE SEARCHING STRATEGY CONSISTED OF 3 COMMANDS
```

5. REFERENCES

5. REFERENCES

1. Nuclear Regulatory Commission, "Reporting of Operating Information - Appendix A: Technical Specifications," Regulatory Guide No. 1.16, Rev. 4 (August 1975).

2. Nuclear Regulatory Commission, Instructions for Preparation of Data Entry Sheets for Licensee Event Report (LER) File, NUREG-0161 (July 1977).

3. Nuclear Regulatory Commission, Licensee Event Report System - Description of System and Guidelines for Reporting, NUREG-1022 (September 1983).

4. R.B. Gallaher, Development of Licensee Event Reports Sequence Coding and Search Procedure, NUREG/CR-1928 (January 1981).

5. R.B. Gallaher, et al., Sequence Coding and Search System Coder's Manual for Licensee Event Reports - Coding Methodology and Code Descriptions, ORNL/NSIC-189, Revision 0, Vols. A and B (March 1984).

6. H.C. Honeck, "The JOSHUA System," Savannah River Laboratories, DB-1380 (April 1975).

7. R.B. Gallaher, et al., Sequence Coding and Search System Coder's Manual for Licensee Event Reports - Code Listings, ORNL/NSIC-189, Revision 0 (March 1984).

6. INDEX OF COMMANDS

6. INDEX OF COMMANDS

NRC FORM 335 (2-84) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)

NUREG/CR-3905
ORNL/NSIC-223

2. TITLE AND SUBTITLE

Sequence Coding and Search System for Licensee Event Reports - User's Guide

3. LEAVE BLANK

4. DATE REPORT COMPLETED

MONTH	YEAR
August	1984

5. AUTHOR(S)

N. M. Greene, G. T. Mays (ORNL)
M. P. Johnson (JBF Associates, Inc.)

6. DATE REPORT ISSUED

MONTH	YEAR
August	1984

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Oak Ridge National Laboratory
Nuclear Operations Analysis Center
Oak Ridge, TN 37831

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER

A9451

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Office for Analysis and Evaluation of Operational Data
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

11a. TYPE OF REPORT

Topical

b. PERIOD COVERED (Inclusive dates)

12. SUPPLEMENTARY NOTES

13. ABSTRACT (200 words or less)

The Nuclear Regulatory Commission's Office for Analysis and Evaluation of Operational Data has developed, through the Nuclear Operations Analysis Center (NOAC) at Oak Ridge National Laboratory (ORNL), a system to aid in the evaluation of the Licensee Event Reports (LERs) submitted by the nuclear power plant utilities. The primary objective of the Sequence Coding and Search System (SCSS) is to reduce the descriptive text of the incident reports to a coded sequence that is both computer-readable and computer-searchable. This system provides a structured format for detailed coding of component, system, and unit effects, as well as personnel errors. The database contains all current LERs submitted by the nuclear power plant utilities after January 1, 1981, and is updated on a continual basis with new LERs, as they are submitted. The database is maintained by NOAC on the IBM-3033 computer system at ORNL. Following a description of SCSS and structure of the database, a tutorial section is provided to acquaint the first-time user with logon procedures and the necessary commands to retrieve, display, and analyze LERs. Each command is subsequently discussed in detail in the fundamental and advanced command sections.

14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS

Licensee Event Report
Database
Reactor, BWR
Reactor, PWR Operating Experience

b. IDENTIFIERS/OPEN-ENDED TERMS

15. AVAILABILITY STATEMENT

Unlimited

16. SECURITY CLASSIFICATION

(This page) Unclassified

(This report) Unclassified

17. NUMBER OF PAGES

18. PRICE

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

120555078877 1 1AN
US NRC
ADM-DIV OF TIDC
POLICY & PUB MGT BR-PDR NUREG
W-501
WASHINGTON DC 20555