SOFTWARE RELEASE NOTICE

1. SRN Number: PA-SRN-260

2. Project Title: TSPAI Code Development – TPA | Project No. 20.01402.762

3. SRN Title: Graphical Post-Processor Version 1.0 for the TPA Code (Version 4.0)

4. Originator/Requestor: Osvaldo Pensado | Date: February 1, 2002

5. Summary of Actions

× Release of new software

☐ Release of modified software:

☐ Enhancements made

☐ Corrections made

☐ Change of access software

☐ Software Retirement

6. Validation Status

☐ Validated

☐ Limited Validation

× Not Validated — Explain: New software, need to reach maturity before validation

7. Persons Authorized Access

Name	Read Only/Read-Write	Addition/Change/Delete
CNWRA technical staff	Read only	
CNWRA TSPAI staff	Read-write	Addition/Change/Delete
NRC staff	Read only	
NRC PA staff	Read-write	Addition/Change/Delete

8. Element Manager Approval: Gordon Wittmeyer Date: 2/1/2002

9. Remarks:

CNWRA Form TOP-6 (09/01)

SOFTWARE SUMMARY FORM

01. Summary Date: February 1, 2002	02. Summary prepared by (Name and phone) Osvaldo Pensado, (210) 522 6084	03. Summary Action: NEW
04. Software Date: Jan 15, 2002	05. Short Title: GPP	
06. Software Title: Graphical Post-Processor Version 1.0 for the TPA Code (Version 4.0)		07. Internal Software ID: none

08. Software Type: ☐ Automated Data System × Computer Program ☐ Subroutine/Module	09. Processing Mode: × Interactive ☐ Batch ☐ Combination	10. Application Area a. General: ☐ Scientific/Engineering ☐ Auxiliary Analyses × Total System PA ☐ Subsystem PA ☐ Other b. Specific: Tool for the graphic display of specific TPA Code output data, and to perform influence analyses

11. Submitting Organization and Address: CNWRA/SwRI 6220 Culebra Road San Antonio, TX 78228	12. Technical Contact(s) and Phone: Joanne Damours Bayesian Systems, Inc. Gaitherburg, MD (301) 987-5400

13. Software Application:

The Graphical Post-Processor supports visualization of specific data generated with the Total-system Performance Assessment (TPA) Version 4.0 code. By plotting relevant TPA output variables, the graphical post-processor is intended as a tool for the quick analysis of the influence of TPA input parameters on relevant variables, and the interrelation among these variables. The graphical post-processor is also intended as a tool for the analysis of uncertainty propagation.

The technical description of the application is presented in the User's Manual. The User's manual includes installation procedures of the graphical post-processor and basic instructions for the graphic display of TPA data, as well as highlights of available features of the graphical post-processor. Appendices contain information useful for software maintenance.

14. Computer Platform Pentium II x86 Family 6 Model 5 Stepping 2 AT/AT Compatible 130,468 KB RAM Make: NetForce Net name: Dakath	15. Computer Operating System: Windows NT Version 4.0	16. Programming Language(s): Java™ 2	17. Number of Source Program Statements: 20,000 lines of code
18. Computer Memory Requirements: At least 96 MB	19. Tape Drives: NA	20. Disk Units: NA	21. Graphics: OpenGL 1.1.2 Java™ 3D 1.1.X

22. Other Operational Requirements
Java™ Runtime Environment 1.3 or Java™ Development Kit 1.3
Colt.jar library, included in the installation CD
Visad.jar library, included in the installation CD
Access to TPA Code output files. A particular set of files must be available for plots to be displayed. The complete set of needed files is explained in the User's manual
Complete installation instructions are available in the User's manual.

23. Software Availability: ☐ Available ☒ Limited ☐ In-House ONLY	24. Documentation Availability: ☐ Available ☒ Preliminary ☐ In-House ONLY

25. Developed by Bayesian Systems, Inc.
Signed by O. Pensado

Software Developer: _____ Date: February 1, 2002

CNWRA Form TOP-4-1 (05/98)

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

Software Title/Name: Graphical Post-Processor (GAP) for TPA Version 4.0

Version: GAP Version 1

Demonstration workstation: "DAKATH", Room A126, Bldg 189

Operating System: Windows NT 4.0

Developer: Bayesian Systems Inc.
Gaithersburg, MD

Software Requirements Description (SRD) [TOP-018, Section 5.3]

SRD Version: SRD dated July 14, 2000, memos dated JAN. 15, 2001 JAN. 15, 2002

SRD Approval Date: July 14, 2000

SRD and any changes thereto reviewed in accordance with QAP-002 requirements?
Yes: ☒ No: ☐ N/A: ☐

Is a Software Change Report(s) (SCR) used for minor modifications (i.e., acquired code), problems or changes to a configured version of software?
Yes: ☐ No: ☒ N/A: ☐

Comments: This is a NEW release. No SCRs At This Time. — 2/7/2002

Software Development Plan (SDP) [TOP-018, Section 5.4]

SDP Version: GAP for TPA Code

SDP (EM) Approval Date: August 17, 2000

The SDP addresses applicable sections of TOP-018, Appendix B, SDP Template?
Yes: ☒ No: ☐ N/A: ☐

Is the waiver (if used) in accordance with specified guidelines?
Yes: ☐ No: ☐ N/A: ☒

Comments:

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

Design and Development [TOP-018, Section 5.5.1 - 5.5.4]

Is code development in accordance with the conventions (i.e., coding conventions)described in the SDP/SCR?

Yes: ☒ No: ☐ N/A: ☐

Module(s) Reviewed: Class GAPApplet
Constance.java

Comments: Dataset.java

Is code internally documented to allow a user to understand the function(s) being performed and to follow the flow of execution of individual routines?

Yes: ☒ No: ☐ N/A: ☐

Module(s) Reviewed: Dataset.java
constance.java

Comments: TPAmin.java

Is development of the code and informal module/subroutine-level testing documented in scientific notebook and/or SCR?

Yes: ☒ No: ☐ N/A: ☐

SCR's and/or Scientific Notebook(s) Reviewed:

Comments: S/N # 375
S/N # 376

Software designed so that individual runs are uniquely identified by date, time, name of software and version?

Yes: ☒ No: ☐ N/A: ☐

Date and Time Displayed: 2002 02 07 5125.21

Name/Version Displayed: GPP1.0

Comments: JPG files are created with a title that includes the GPP version, the date, and the time of creation. The same applies to comma-separated value (CSV) files.

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

Medium and Header Documentation [TOP-018, Section 5.5.6]

A program title block of main program contains: Program Title, Customer Name, Customer Office/Division, Customer Contact(s), Customer Phone Number, Associated Documentation, Software Developer and Phone Number, Date, and Disclaimer Notice?

Yes: ☒ No: ☐ N/A: ☐

Comments: See printed page in this documentation.

Source code module headers contain: Program Name, Client Name, Contract reference, Revision Number, Revision History, and Reference to SRD/SCR requirement(s)?

Yes: ☒ No: ☐ N/A: ☐

Module(s) Reviewed: TPARun.java ✓

Comments: JAVA is an object-oriented code, so it does not have "modules," but it has "objects." Each "object" has a header as required. SE

The physical labeling of software medium (tapes, disks, etc.) contains: Program Name, Module/Name/Title, Module Revision, File type (ASCII, OBJ, EXE), Recording Date, and Operating System(s)?

Yes: ☒ No: ☐ N/A: ☐

Comments: See the CD in this folder. SE

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

Code Reviews [TOP-018, Section 5.5.6]

Are code reviews (if implemented) documented in a scientific notebook or in another format that allows others to understand the code review process and results?

Yes: ☐ No: ☐ N/A: ☒

Documented in Scientific Notebook No.: N/A

Comments:

Acceptance and Installation Testing [TOP-018, Section 5.6]

Does *acceptance testing* demonstrate whether or not requirements in the SRD and/or SCR(s) have been fulfilled?

Yes: ☒ No: ☐ N/A: ☐

Has *acceptance testing* been conducted for each intended computer platform and operating system?

Yes: ☒ No: ☐ N/A: ☐

Computer Platforms: PC Operating Systems: Windows NT 4.0

Location of Acceptance Test Results: Electronic Scientific Notebook No. 376-EZ

Comments:

See pages printed from O. Pensado's S/N 376-EZ.

Has *installation testing* been conducted for each intended computer platform and operating system?

Yes: ☒ No: ☐ N/A: ☐

Computer Platforms: PC Operating Systems: Windows NT 4.0

Location of Acceptance Test Results: Electronic Scientific Notebook No. 376-EZ.

Comments:

S. Mohyee also performed installation testing when doing QAP-002 review of The User's Manual.

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

User Documentation [TOP-018, Section 5.5.7]

Is there a Users' Manual for the software and is it up-to-date? Yes: ☒ No: ☐ N/A: ☐

User's Manual Version and Date: Users Guide To Version 1.0 of The GAP for TPA V4.0. February 2002

Comments: The GAP V1.0 code & Users Guide is delivered to NRC Feb. 2002.

Are there basic instructions for the *installation* and *use* of the software? Yes: ☒ No: ☐ N/A: ☐

Location of Instructions: Users Guide To Version 1.0 of The GAP.

Comments: See Users Guide in This File.

Configuration Control [TOP-018, Section 5.7, 5.9.3]

Is the Software Summary Form (Form TOP-4-1) completed and signed? Yes: ☒ No: ☐ N/A: ☐

Date of Approval: Feb. 1, 2002

Is the list of files attached to the Software Summary Form complete and accurate? Yes: ☒ No: ☐ N/A: ☒

Comments:

Is the source code available or, is the executable code available in the case of (acquired/commercial codes)? Yes: ☒ No: ☐ N/A: ☐

Location of Source Code: In This Folder

Comments:

Have all the script/make files and executable files been submitted to the Software Custodian? Yes: ☒ No: ☐ N/A: ☐

Location of script/make files: On Source Code CD in This Folder.

Comments:

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
DESIGN VERIFICATION REPORT
FOR
→ *DEVELOPED OR ACQUIRED TO BE MODIFIED SOFTWARE* ←

Software Release [TOP-018, Section 5.9]

Upon acceptance of the software as verified above, has a Software Release Notice (SRN), Form TOP-6 been issued and does the version number of the software match the documentation?

Yes: ☒ No: ☐ N/A: ☐

SRN Number: PA-SRN-260

Comments: See SRD in this folder.

Software Validation [TOP-018, Section 5.10]

Has a Software Validation Test Plan (SVTP) been prepared for the range of application of the software?

Yes: ☐ No: ☒ N/A: ☐

Version and Date of SVTP: Not as of 2/2002

Date Reviewed and Approved via QAP-002: _____

Comments: This is newly developed software and it will be presented to the NRC in 2/2002. Validation will be by NRC direction. 2/7/2002

Has a Software Validation Test Report (SVTR) been prepared that documents the results of the validation cases, interpretation of the results, and determination if the software has been validated?

Yes: ☐ No: ☒ N/A: ☐

Version and Date of SVTR: _____

Date Reviewed and Approved via QAP-002: _____

Comments: See above comments.

Additional Comments:

Signing by Bayesian Systems, Inc.

2/8/2002
Software Developer/Date

Software Custodian/Date 2/8/2002

List of files of the CD containing the Graphical Post-Processor **Version 1.0** for the TPA code.

F:\compileJava.bat
F:\gpp
F:\java
F:\launch.bat
F:\libraries
F:\NuclNames.txt
F:\readme.txt
F:\Repository Performance Visualization.pdf
F:\Repository Performance Visualization.wpd
F:\TPAData
F:\gpp\02-02-07
F:\gpp\api-doc
F:\gpp\02-02-07\com
F:\gpp\02-02-07\gpp.properties
F:\gpp\02-02-07\logCriteria.properties
F:\gpp\02-02-07\log.properties
F:\gpp\02-02-07\net
F:\gpp\02-02-07\outputfiles.properties
F:\gpp\02-02-07\ptolemy
F:\gpp\02-02-07\variables-orig.properties
F:\gpp\02-02-07\variables.properties
F:\gpp\02-02-07\variable_abbrevs.properties
F:\gpp\02-02-07\com\bayes
F:\gpp\02-02-07\com\microstar
F:\gpp\02-02-07\com\bayes\gpp
F:\gpp\02-02-07\com\bayes\gpp\a
F:\gpp\02-02-07\com\bayes\gpp\a1
F:\gpp\02-02-07\com\bayes\gpp\a1p
F:\gpp\02-02-07\com\bayes\gpp\a1v
F:\gpp\02-02-07\com\bayes\gpp\a3
F:\gpp\02-02-07\com\bayes\gpp\a4
F:\gpp\02-02-07\com\bayes\gpp\a5
F:\gpp\02-02-07\com\bayes\gpp\a6
F:\gpp\02-02-07\com\bayes\gpp\aa
F:\gpp\02-02-07\com\bayes\gpp\Constants.class
F:\gpp\02-02-07\com\bayes\gpp\Constants.java
F:\gpp\02-02-07\com\bayes\gpp\DataSet.class
F:\gpp\02-02-07\com\bayes\gpp\DataSet.java
F:\gpp\02-02-07\com\bayes\gpp\FArray.class
F:\gpp\02-02-07\com\bayes\gpp\FArray.java
F:\gpp\02-02-07\com\bayes\gpp\FieldPanel$ListSelectionHandler.class
F:\gpp\02-02-07\com\bayes\gpp\FieldPanel.java
F:\gpp\02-02-07\com\bayes\gpp\FieldPanel.class
F:\gpp\02-02-07\com\bayes\gpp\Field.class
F:\gpp\02-02-07\com\bayes\gpp\Field.java
F:\gpp\02-02-07\com\bayes\gpp\Fmt.class
F:\gpp\02-02-07\com\bayes\gpp\Fmt.java
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$5.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$6.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$7.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$8.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$FileNameGenerator.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$RadioListenerWindow.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$RadioListenerTPA.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$SnapThread.class

F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$1.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet.html
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet.java
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$2.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$3.class
F:\gpp\02-02-07\com\bayes\gpp\GPPApplet$4.class
F:\gpp\02-02-07\com\bayes\gpp\Legend$1.class
F:\gpp\02-02-07\com\bayes\gpp\Legend.java
F:\gpp\02-02-07\com\bayes\gpp\Legend.class
F:\gpp\02-02-07\com\bayes\gpp\Lines.class
F:\gpp\02-02-07\com\bayes\gpp\Lines.java
F:\gpp\02-02-07\com\bayes\gpp\MultiFamilyTimeBasedPlot.class
F:\gpp\02-02-07\com\bayes\gpp\MultiFamilyTimeBasedPlot.java
F:\gpp\02-02-07\com\bayes\gpp\Parameter.class
F:\gpp\02-02-07\com\bayes\gpp\Parameter.java
F:\gpp\02-02-07\com\bayes\gpp\ParameterReference.class
F:\gpp\02-02-07\com\bayes\gpp\ParameterReference.java
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$VarListSelectionHandler.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$RadioListenerField.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$PcListSelectionHandler.class
F:\gpp\02-02-07\com\bayes\gpp\Place.class
F:\gpp\02-02-07\com\bayes\gpp\Place.java
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$ParamListSelectionHandler.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$6.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$5.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$1.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame.java
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$2.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$3.class
F:\gpp\02-02-07\com\bayes\gpp\PlotControlFrame$4.class
F:\gpp\02-02-07\com\bayes\gpp\PlotSpecification.class
F:\gpp\02-02-07\com\bayes\gpp\PlotSpecification.java
F:\gpp\02-02-07\com\bayes\gpp\PlottableData.class
F:\gpp\02-02-07\com\bayes\gpp\PlottableData.java
F:\gpp\02-02-07\com\bayes\gpp\PlottableRawData.class
F:\gpp\02-02-07\com\bayes\gpp\PlottableRawData.java
F:\gpp\02-02-07\com\bayes\gpp\PlottableRegularizedData.class
F:\gpp\02-02-07\com\bayes\gpp\PlottableRegularizedData.java
F:\gpp\02-02-07\com\bayes\gpp\PlotterBase$FileNameGenerator.class
F:\gpp\02-02-07\com\bayes\gpp\PlotterBase.java
F:\gpp\02-02-07\com\bayes\gpp\PlotterBase.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog$5.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2$5.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2$1.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2$2.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2$3.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2$4.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog$1.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog.java
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog$2.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog2.java
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog$3.class
F:\gpp\02-02-07\com\bayes\gpp\PromptDialog$4.class
F:\gpp\02-02-07\com\bayes\gpp\PtolemaicInput.class
F:\gpp\02-02-07\com\bayes\gpp\PtolemaicInput.java
F:\gpp\02-02-07\com\bayes\gpp\RegularizedTPAData.class
F:\gpp\02-02-07\com\bayes\gpp\RegularizedTPAData.java

F:\gpp\02-02-07\com\bayes\gpp\TPAData.class
F:\gpp\02-02-07\com\bayes\gpp\TPAData.java
F:\gpp\02-02-07\com\bayes\gpp\TPADCAGWInputValuesOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPADCAGWInputValuesOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPADCFOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPADCFOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPANoRealizationsOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPANoRealizationsOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPANuclideOnlyOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPANuclideOnlyOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPAOutput.class
F:\gpp\02-02-07\com\bayes\gpp\TPAOutput.java
F:\gpp\02-02-07\com\bayes\gpp\TPAOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPAOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPAParameter.class
F:\gpp\02-02-07\com\bayes\gpp\TPAParameter.java
F:\gpp\02-02-07\com\bayes\gpp\TPAPeakDosePeakTimeOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPAPeakDosePeakTimeOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPARun.class
F:\gpp\02-02-07\com\bayes\gpp\TPARun.java
F:\gpp\02-02-07\com\bayes\gpp\TPASinglePointByNuclideOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPASinglePointByNuclideOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPASparseTimeOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPASparseTimeOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPASubareaByTimeOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPASubareaByTimeOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\TPASubareaNuclideByTimeOutputLoadStrategy.class
F:\gpp\02-02-07\com\bayes\gpp\TPASubareaNuclideByTimeOutputLoadStrategy.java
F:\gpp\02-02-07\com\bayes\gpp\Utils.class
F:\gpp\02-02-07\com\bayes\gpp\Utils.java
F:\gpp\02-02-07\com\bayes\gpp\Variable.class
F:\gpp\02-02-07\com\bayes\gpp\Variable.java
F:\gpp\02-02-07\com\bayes\gpp\ViewControlFrame$1.class
F:\gpp\02-02-07\com\bayes\gpp\ViewControlFrame.java
F:\gpp\02-02-07\com\bayes\gpp\ViewControlFrame$2.class
F:\gpp\02-02-07\com\bayes\gpp\ViewControlFrame$3.class
F:\gpp\02-02-07\com\bayes\gpp\ViewControlFrame.class
F:\gpp\02-02-07\com\bayes\gpp\View.class
F:\gpp\02-02-07\com\bayes\gpp\View.java
F:\gpp\02-02-07\com\microstar\xml
F:\gpp\02-02-07\com\microstar\xml\demo
F:\gpp\02-02-07\com\microstar\xml\HandlerBase.class
F:\gpp\02-02-07\com\microstar\xml\SAXDriver.class
F:\gpp\02-02-07\com\microstar\xml\XmlException.class
F:\gpp\02-02-07\com\microstar\xml\XmlHandler.class
F:\gpp\02-02-07\com\microstar\xml\XmlParser.class
F:\gpp\02-02-07\com\microstar\xml\demo\DtdDemo.class
F:\gpp\02-02-07\com\microstar\xml\demo\EventDemo.class
F:\gpp\02-02-07\com\microstar\xml\demo\ReaderDemo.class
F:\gpp\02-02-07\com\microstar\xml\demo\StreamDemo.class
F:\gpp\02-02-07\com\microstar\xml\demo\TimerDemo.class
F:\gpp\02-02-07\com\microstar\xml\demo\XmlApp.class
F:\gpp\02-02-07\com\microstar\xml\demo\XmlApplet.class
F:\gpp\02-02-07\net\mirabile
F:\gpp\02-02-07\net\mirabile\logger
F:\gpp\02-02-07\net\mirabile\Util.class
F:\gpp\02-02-07\net\mirabile\Util.java
F:\gpp\02-02-07\net\mirabile\logger\Appender.class
F:\gpp\02-02-07\net\mirabile\logger\Appender.java
F:\gpp\02-02-07\net\mirabile\logger\Category$CategoryNode.class

F:\gpp\02-02-07\net\mirabile\logger\Category.java
F:\gpp\02-02-07\net\mirabile\logger\Category$ProvisionNode.class
F:\gpp\02-02-07\net\mirabile\logger\Category.class
F:\gpp\02-02-07\net\mirabile\logger\FileAppender.class
F:\gpp\02-02-07\net\mirabile\logger\FileAppender.java
F:\gpp\02-02-07\net\mirabile\logger\ILog.class
F:\gpp\02-02-07\net\mirabile\logger\ILog.java
F:\gpp\02-02-07\net\mirabile\logger\Layout.class
F:\gpp\02-02-07\net\mirabile\logger\Layout.java
F:\gpp\02-02-07\net\mirabile\logger\Log.class
F:\gpp\02-02-07\net\mirabile\logger\Log.java
F:\gpp\02-02-07\net\mirabile\logger\MyLayout.class
F:\gpp\02-02-07\net\mirabile\logger\MyLayout.java
F:\gpp\02-02-07\net\mirabile\logger\NDC$DiagnosticContext.class
F:\gpp\02-02-07\net\mirabile\logger\NDC.class
F:\gpp\02-02-07\net\mirabile\logger\NDC.java
F:\gpp\02-02-07\net\mirabile\logger\NOPLog.class
F:\gpp\02-02-07\net\mirabile\logger\NOPLog.java
F:\gpp\02-02-07\net\mirabile\logger\ObjectProperties.class
F:\gpp\02-02-07\net\mirabile\logger\ObjectProperties.java
F:\gpp\02-02-07\net\mirabile\logger\Priority.class
F:\gpp\02-02-07\net\mirabile\logger\Priority.java
F:\gpp\02-02-07\net\mirabile\logger\QuietWriter.class
F:\gpp\02-02-07\net\mirabile\logger\QuietWriter.java
F:\gpp\02-02-07\net\mirabile\logger\SimpleLayout.class
F:\gpp\02-02-07\net\mirabile\logger\SimpleLayout.java
F:\gpp\02-02-07\net\mirabile\logger\Tracer.class
F:\gpp\02-02-07\net\mirabile\logger\Tracer.java
F:\gpp\02-02-07\net\mirabile\logger\TracerPrintWriter.class
F:\gpp\02-02-07\net\mirabile\logger\TracerPrintWriter.java
F:\gpp\02-02-07\ptolemy\gui
F:\gpp\02-02-07\ptolemy\plot
F:\gpp\02-02-07\ptolemy\gui\BasicJApplet.class
F:\gpp\02-02-07\ptolemy\gui\BasicJApplet.java
F:\gpp\02-02-07\ptolemy\gui\ComponentDialog$1.class
F:\gpp\02-02-07\ptolemy\gui\ComponentDialog.java
F:\gpp\02-02-07\ptolemy\gui\ComponentDialog.class
F:\gpp\02-02-07\ptolemy\gui\demo
F:\gpp\02-02-07\ptolemy\gui\gui.jar
F:\gpp\02-02-07\ptolemy\gui\makefile
F:\gpp\02-02-07\ptolemy\gui\Message$1.class
F:\gpp\02-02-07\ptolemy\gui\Message.java
F:\gpp\02-02-07\ptolemy\gui\Message$2.class
F:\gpp\02-02-07\ptolemy\gui\Message.class
F:\gpp\02-02-07\ptolemy\gui\Query$QueryActionListener.class
F:\gpp\02-02-07\ptolemy\gui\Query$QueryFocusListener.class
F:\gpp\02-02-07\ptolemy\gui\Query$QueryItemListener.class
F:\gpp\02-02-07\ptolemy\gui\Query$SliderListener.class
F:\gpp\02-02-07\ptolemy\gui\QueryListener.class
F:\gpp\02-02-07\ptolemy\gui\QueryListener.java
F:\gpp\02-02-07\ptolemy\gui\Query.class
F:\gpp\02-02-07\ptolemy\gui\Query.java
F:\gpp\02-02-07\ptolemy\gui\StatusBar.class
F:\gpp\02-02-07\ptolemy\gui\StatusBar.java
F:\gpp\02-02-07\ptolemy\gui\demo\makefile
F:\gpp\02-02-07\ptolemy\gui\demo\QueryApplet.htm
F:\gpp\02-02-07\ptolemy\gui\demo\QueryApplet.java
F:\gpp\02-02-07\ptolemy\gui\demo\ShowRawDocumentApplet.htm
F:\gpp\02-02-07\ptolemy\gui\demo\ShowRawDocumentApplet.java
F:\gpp\02-02-07\ptolemy\plot\CmdLineArgException.class

F:\gpp\02-02-07\ptolemy\plot\CmdLineArgException.java
F:\gpp\02-02-07\ptolemy\plot\EditablePlot$EditMouseListener.class
F:\gpp\02-02-07\ptolemy\plot\EditablePlot.java
F:\gpp\02-02-07\ptolemy\plot\EditablePlot$ModifyListener.class
F:\gpp\02-02-07\ptolemy\plot\EditablePlot$UndoListener.class
F:\gpp\02-02-07\ptolemy\plot\EditablePlot.class
F:\gpp\02-02-07\ptolemy\plot\EditListener.class
F:\gpp\02-02-07\ptolemy\plot\EditListener.java
F:\gpp\02-02-07\ptolemy\plot\EPSGraphics.class
F:\gpp\02-02-07\ptolemy\plot\EPSGraphics.java
F:\gpp\02-02-07\ptolemy\plot\PlotFrame.class
F:\gpp\02-02-07\ptolemy\plot\PlotFrame$SpecialMenuListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotBox.class
F:\gpp\02-02-07\ptolemy\plot\Plot.jbx
F:\gpp\02-02-07\ptolemy\plot\Plot$$3.class
F:\gpp\02-02-07\ptolemy\plot\Plot$1.class
F:\gpp\02-02-07\ptolemy\plot\Plot$2.class
F:\gpp\02-02-07\ptolemy\plot\Plot$Format.class
F:\gpp\02-02-07\ptolemy\plot\PlotApplet.class
F:\gpp\02-02-07\ptolemy\plot\PlotApplet.java
F:\gpp\02-02-07\ptolemy\plot\PlotApplication$1.class
F:\gpp\02-02-07\ptolemy\plot\PlotApplication.java
F:\gpp\02-02-07\ptolemy\plot\PlotApplication.class
F:\gpp\02-02-07\ptolemy\plot\PlotBox$CommandListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotBox.java
F:\gpp\02-02-07\ptolemy\plot\PlotBox$DragListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotBox$FillButtonListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotBox$ZoomListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotFrame$1.class
F:\gpp\02-02-07\ptolemy\plot\PlotFrame.java
F:\gpp\02-02-07\ptolemy\plot\PlotFrame$2.class
F:\gpp\02-02-07\ptolemy\plot\PlotFrame$FileMenuListener.class
F:\gpp\02-02-07\ptolemy\plot\PlotFrame$FormatListener.class
F:\gpp\02-02-07\ptolemy\plot\plotml
F:\gpp\02-02-07\ptolemy\plot\plotml.dtd
F:\gpp\02-02-07\ptolemy\plot\PlotPoint.class
F:\gpp\02-02-07\ptolemy\plot\PlotPoint.java
F:\gpp\02-02-07\ptolemy\plot\Plot.class
F:\gpp\02-02-07\ptolemy\plot\Plot.java
F:\gpp\02-02-07\ptolemy\plot\whisker
F:\gpp\02-02-07\ptolemy\plot\whisker.dtd
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplication.class
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplet$$1.class
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplet.java
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplet$ChoiceListener.class
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplication.java
F:\gpp\02-02-07\ptolemy\plot\plotml\EditablePlotMLApplet.class
F:\gpp\02-02-
07\ptolemy\plot\plotml\EditablePlotMLApplication$SelectListener.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotBoxMLParser.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotBoxMLParser.java
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLApplet.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLApplet.java
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLApplication.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLApplication.java
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLFrame.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLFrame.java
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLParser.class
F:\gpp\02-02-07\ptolemy\plot\plotml\PlotMLParser.java
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker.bat

F:\gpp\02-02-07\ptolemy\plot\whisker\whisker.dtd
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker.htm
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker.xml
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker-DOUG.htm
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker-A.xml
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker-B.xml
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker-C.xml
F:\gpp\02-02-07\ptolemy\plot\whisker\whisker-SAVE.xml
F:\gpp\api-doc\allclasses-frame.html
F:\gpp\api-doc\com
F:\gpp\api-doc\deprecated-list.html
F:\gpp\api-doc\generate_javadoc.bat
F:\gpp\api-doc\help-doc.html
F:\gpp\api-doc\index-all.html
F:\gpp\api-doc\index.html
F:\gpp\api-doc\overview-tree.html
F:\gpp\api-doc\package-list
F:\gpp\api-doc\packages.html
F:\gpp\api-doc\serialized-form.html
F:\gpp\api-doc\setclassp.bat
F:\gpp\api-doc\stylesheet.css
F:\gpp\api-doc\com\bayes
F:\gpp\api-doc\com\bayes\gpp
F:\gpp\api-doc\com\bayes\gpp\Constants.html
F:\gpp\api-doc\com\bayes\gpp\DataSet.html
F:\gpp\api-doc\com\bayes\gpp\FArray.html
F:\gpp\api-doc\com\bayes\gpp\FieldPanel.html
F:\gpp\api-doc\com\bayes\gpp\FieldPanel.ListSelectionHandler.html
F:\gpp\api-doc\com\bayes\gpp\Field.html
F:\gpp\api-doc\com\bayes\gpp\Fmt.html
F:\gpp\api-doc\com\bayes\gpp\GPPApplet.SnapThread.html
F:\gpp\api-doc\com\bayes\gpp\GPPApplet.FileNameGenerator.html
F:\gpp\api-doc\com\bayes\gpp\GPPApplet.html
F:\gpp\api-doc\com\bayes\gpp\GPPApplet.RadioListenerTPA.html
F:\gpp\api-doc\com\bayes\gpp\GPPApplet.RadioListenerWindow.html
F:\gpp\api-doc\com\bayes\gpp\Legend.html
F:\gpp\api-doc\com\bayes\gpp\Lines.html
F:\gpp\api-doc\com\bayes\gpp\MultiFamilyTimeBasedPlot.html
F:\gpp\api-doc\com\bayes\gpp\package-frame.html
F:\gpp\api-doc\com\bayes\gpp\package-summary.html
F:\gpp\api-doc\com\bayes\gpp\package-tree.html
F:\gpp\api-doc\com\bayes\gpp\Parameter.html
F:\gpp\api-doc\com\bayes\gpp\ParameterReference.html
F:\gpp\api-doc\com\bayes\gpp\PlotControlFrame.VarListSelectionHandler.html
F:\gpp\api-doc\com\bayes\gpp\Place.html
F:\gpp\api-doc\com\bayes\gpp\PlotControlFrame.html
F:\gpp\api-doc\com\bayes\gpp\PlotControlFrame.ParamListSelectionHandler.html
F:\gpp\api-doc\com\bayes\gpp\PlotControlFrame.PcListSelectionHandler.html
F:\gpp\api-doc\com\bayes\gpp\PlotControlFrame.RadioListenerField.html
F:\gpp\api-doc\com\bayes\gpp\PlotSpecification.html
F:\gpp\api-doc\com\bayes\gpp\PlottableData.html
F:\gpp\api-doc\com\bayes\gpp\PlottableRawData.html
F:\gpp\api-doc\com\bayes\gpp\PlottableRegularizedData.html
F:\gpp\api-doc\com\bayes\gpp\PlotterBase.FileNameGenerator.html
F:\gpp\api-doc\com\bayes\gpp\PlotterBase.html
F:\gpp\api-doc\com\bayes\gpp\PromptDialog.html
F:\gpp\api-doc\com\bayes\gpp\PromptDialog2.html
F:\gpp\api-doc\com\bayes\gpp\PtolemaicInput.html
F:\gpp\api-doc\com\bayes\gpp\RegularizedTPAData.html
F:\gpp\api-doc\com\bayes\gpp\TPAData.html

F:\gpp\api-doc\com\bayes\gpp\TPADCAGWInputValuesOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPADCFOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPANoRealizationsOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPANuclideOnlyOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPAOutput.html
F:\gpp\api-doc\com\bayes\gpp\TPAOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPAParameter.html
F:\gpp\api-doc\com\bayes\gpp\TPAPeakDosePeakTimeOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPARun.html
F:\gpp\api-doc\com\bayes\gpp\TPASinglePointByNuclideOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPASparseTimeOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPASubareaByTimeOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\TPASubareaNuclideByTimeOutputLoadStrategy.html
F:\gpp\api-doc\com\bayes\gpp\Utils.html
F:\gpp\api-doc\com\bayes\gpp\Variable.html
F:\gpp\api-doc\com\bayes\gpp\ViewControlFrame.html
F:\gpp\api-doc\com\bayes\gpp\View.html
F:\java\jdk
F:\java\jre
F:\java\jdk\download
F:\java\jdk\download\J2SDK1_3_0-DOC.ZIP
F:\java\jdk\download\JDK130.EXE
F:\java\jdk\download\tutorial.zip
F:\java\jre\download
F:\java\jre\j2-datasheet.htm
F:\java\jre\j2re1_3_0_license.txt
F:\java\jre\j2sdk13-features.htm
F:\java\jre\java_sun_com Downloading FAQ.htm
F:\java\jre\jre-130-readme.txt
F:\java\jre\jre-130-changes.txt
F:\java\jre\jre-inst-notes.htm
F:\java\jre\jre-troubleshooting.htm
F:\java\jre\Terms&conditions of license&export for Java(TM) 2 R.htm
F:\java\jre\download\j2re1_3_0-win.exe
F:\libraries\colt
F:\libraries\java3d
F:\libraries\opengl
F:\libraries\visad
F:\libraries\colt\colt.jar
F:\libraries\colt\Colt-home_files
F:\libraries\colt\colt-home.htm
F:\libraries\colt\download
F:\libraries\colt\license.htm
F:\libraries\colt\src.zip
F:\libraries\colt\this_is_version_101
F:\libraries\colt\Colt-home_files\4stars.gif
F:\libraries\colt\Colt-home_files\counter.gif
F:\libraries\colt\Colt-home_files\counter9.xbm
F:\libraries\colt\Colt-home_files\Jarstop5.gif
F:\libraries\colt\Colt-home_files\logo2.gif
F:\libraries\colt\Colt-home_files\stylesheet.css
F:\libraries\colt\download\colt.jar
F:\libraries\colt\download\colt1.0.1.zip
F:\libraries\java3d\java3d1_1_2-win32-opengl-sdk.exe
F:\libraries\opengl\OPENGL95.EXE
F:\libraries\opengl\origin.txt
F:\libraries\visad\download
F:\libraries\visad\top-doc
F:\libraries\visad\visad.jar
F:\libraries\visad\download\v-doc.jar

F:\libraries\visad\download\v-eg.jar
F:\libraries\visad\download\v-mail.zip
F:\libraries\visad\download\v-src.jar
F:\libraries\visad\download\v-tutor.jar
F:\libraries\visad\download\visad.jar
F:\libraries\visad\top-doc\faq.htm
F:\libraries\visad\top-doc\GNU-Public-License.txt
F:\libraries\visad\top-doc\guide.htm
F:\libraries\visad\top-doc\jama.htm
F:\libraries\visad\top-doc\README.txt
F:\libraries\visad\top-doc\README_ss.txt
F:\libraries\visad\top-doc\rme-brow.htm
F:\libraries\visad\top-doc\rme-py.txt
F:\libraries\visad\top-doc\v-home.htm
F:\libraries\visad\top-doc\v-home_files
F:\libraries\visad\top-doc\visad-ss.htm
F:\libraries\visad\top-doc\visad-ss_files
F:\libraries\visad\top-doc\v-home_files\galaxy_sphere.gif
F:\libraries\visad\top-doc\v-home_files\j2j.jpg
F:\libraries\visad\top-doc\v-home_files\visad_fits.gif
F:\libraries\visad\top-doc\visad-ss_files\editmap.jpg
F:\libraries\visad\top-doc\visad-ss_files\lilmini.jpg
F:\libraries\visad\top-doc\visad-ss_files\lilsnap1.jpg
F:\libraries\visad\top-doc\visad-ss_files\lilsnap2.jpg
F:\TPAData\cumrel.res
F:\TPAData\dcagw.ech
F:\TPAData\dcf.cum
F:\TPAData\ebsfail.rlt
F:\TPAData\ebsflo.dat
F:\TPAData\ebsrel.rlt
F:\TPAData\gwpkdos.res
F:\TPAData\infilper.res
F:\TPAData\nfenv.rlt
F:\TPAData\samplpar.hdr
F:\TPAData\samplpar.res
F:\TPAData\szft.rlt
F:\TPAData\totdose.res
F:\TPAData\tpa.inp
F:\TPAData\uzflow.rlt
F:\TPAData\uzft.rlt

TPAmn.java

```
/*
 * Title:        Graphical Post-Processor for TPA Version 4.0 Code<p>
 * Description:  Graphical Post-Processor for TPA Version 4.0 Code, v.
1.0
 * Bayesian Systems, Gaithersburg, MD
 * For CNWRA, San Antonio, TX<p>
 * <p>
 * Company:      Bayesian Systems, Inc.<p>
 * @author John Emmerling
 * @version 1.0
 */
package com.bayes.gpp;

import java.io.LineNumberReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.EOFException;
import java.util.Hashtable;

/**
 * <table>
 * <tr>
 * <td valign=top>Program Name:</td><td>        Graphical Post-Processor
for the TPA Code Version 4.0</td>
 * </tr>
 * <tr>
 * <td valign=top>File Date:</td><td>           2001-09-03</td>
 * </tr>
 * <tr>
 * <td valign=top>Release Version:</td><td>     1.0</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Client Name:</td><td><table>         <tr><td>U.S.
NRC</td></tr>
 *                      <tr><td>U.S. Nuclear Regulatory
Commission</td></tr>
 *                      <tr><td>NRC Office of Nuclear Material Safety
and Safeguards</td></tr>
 *                      <tr><td>Division of Waste
Management</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Contract Number:</td><td>     NRC 02-97-009</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>NRC Contact:</td><td>         James Firth (301) 415-
6628</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>CNWRA Contact:</td><td>       <table><tr><td>Osvaldo
Pensado (210) 522-6084</td></tr>
 *                      <tr><td>Center for Nuclear Waste Regulatory
Analyses</td></tr>
 *                      <tr><td>San Antonio, Texas 78238-5166</td></tr>
```

"object header"

```
 *                        <tr><td>opensado@swri.edu</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Author:</td><td>                John Emmerling</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>Bayesian Systems, Inc., Gaithersburg,
MD</td>
 * </tr>
 * <tr>
 * <td valign=top>Contact:</td><td> <table><tr><td>Joanne Damours (301)
987-5400</td></tr>
 *
<tr><td>joanne@bayes.com</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Documentation:</td><td>       A Graphical Post-
Processor for TPA Code Version 4.0, User's Guide. Version 1.0</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>                        Center for Nuclear
Waste Regulatory Analyses</td>
 * </tr>
 * <tr>
 *
 *
 * <td valign=top>NUREG-Series Designator:</td><td> N/A</td>
 * </tr>
 * </table>
 *
 *
 *
<center><h2>D I S C L A I M E&nb
sp;R</h2></center>
 *
 *
 *     <p>&quot;This computer code/material was prepared as an account
of work
 * performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA)
for
 * the Division of Waste Management of the Nuclear Regulatory Commission
(NRC),
 * an independent agency of the United States Government. Neither the
 * developer(s) of the code nor any of their sponsors make any warranty,
 * expressed or implied, or assume any legal liability or responsibility
for
 * the accuracy, completeness, or usefulness of any information,
apparatus,
 * product or process disclosed, or represent that its use would not
infringe
 * on privately-owned rights.&quot;</p>
 *
 *     <p>&quot;In no event unless required by applicable law will the
sponsors or
 * those who have written or modified this code, be liable for damages,
 * including any lost profits, lost monies, or other special, incidental
or
 * consequential damages arising out of the use or inability to use the
program
```

```
 * (including but not limited to loss of data or data being rendered
inaccurate
 * or losses sustained by third parties or a failure of the program to
operate
 * with other programs), even if you have been advised of the
possibility of
 * such damages or for any claim by any other party.&amp;quot;</p>
 *
 * <HR>
 *
 * <p>The TPARun class encapsulates all the information required
 * about an individual TPA Run.</p>
 *
 * @since 1.0
 * @author John Emmerling
 * @version 1.0
 */
public class TPARun {

  /** The set of all instances of this class
   */
  private static Hashtable AllRuns = new Hashtable();
  /** Denotes TPA run currently selected by the user.
   */
  private static TPARun selectedRun;

  /** hard-coded name of file from which information regarding
   * nuclides may be obtained.
   */
  private static final String NUCLIDES_FILE = "nuclides.dat";

  /** The path to TPA output associated with the run.
   */
  String path = "E:/GPP/tpa/testruns/01/";

  /** Set of data points used to construct logarithmic time scale.
   * Based on time points read from the Water Entering Drift
   * variable.
   */
  private PlottableRawData logTimes;
  /** Set of data points used to construct linear time scale.
   * Based on time points read from the Water Entering Drift
   * variable.
   */
  private PlottableRawData linearTimes;
  /** Number of realizations in TPA Run, not withstanding override.
   */
  private int numRealizations = 0;
  /** Number of nuclides in TPA Run.
   */
  private int numNuclides = 0;
  /** Number of subareas in TPA Run.
   */
  private int numSubareas = 16;

  /*
  static {
    TPARun r = new TPARun();
    AllRuns.add(r);
    selectedRun = r;
  }
```

```
   */

   /** Factory method for creating a TPA run.
    *
    * @param path the path to TPA output associated with the run
    * @param numSubareas number of subareas
    * @param numNuclides number of nuclides
    */
   static void createTPARun(String path,int numSubareas, int numNuclides)
{
     TPARun r = new TPARun();
     r.setPath(path);
     r.setNumSubareas(numSubareas);
     r.setNumNuclides(numNuclides);
     AllRuns.put(path,r);
     selectedRun = r;
     selectedRun.initialize();
   }

   // To be called on application start-up
   /** Initialization method for the class.
    */
   static void initTPARun() {
     // temporary
     selectedRun.initialize();
   }

   /** Constructor.
    */
   public TPARun() {
   }

   /** Accessor method for path
    *
    * @return path
    */
   public String getPath () {
     return path;
   }

   /** Mutator method for path
    *
    * @param path path
    */
   public void setPath(String path) {
     this.path = path;
   }

   /** echos the path
    *
    * @return path
    */
   public String toString () {
     return getPath();
   }

   // public FArray getTimes () {
   /** Accessor method for logTimes.  Returns a trivial result if
    * logTimes not yet defined.
    *
    * @return log times
```

```
 */
public PlottableData getLogTimes () {
  if (logTimes == null) {
    return new PlottableRawData(1,1,1,"bogus","bogus","bogus");
  } else {
    return logTimes;
  }
}

/** Accessor method for linearTimes.  Returns a trivial result if
 * linearTimes not yet defined.
 *
 * @return linear times
 */
public PlottableData getLinearTimes () {
  if (linearTimes == null) {
    return new PlottableRawData(1,1,1,"bogus","bogus","bogus");
  } else {
    return linearTimes;
  }
}

/** Accessor method for selectedRun
 *
 * @return selected run
 */
public static TPARun getSelectedRun () {
  return selectedRun;
}

/** determines number of time points based on length of logTimes.
 *
 * @return number of time points
 */
public static int getNumTimes() {
  return selectedRun.getLogTimes().nRows();
}

/** Cause the run having the specified path to become the
 * new selected run.
 *
 * @param path path of run to be selected
 */
public static void selectRun(String path) {
  selectedRun = (TPARun)AllRuns.get(path);
}

/** Accessor for AllRuns
 *
 * @return all runs as Hashtable
 */
public static Hashtable getAllRuns() {
  return AllRuns;
}

/** Gets the "official" number of realizations for the selected run,
 * taking override into account.
 *
 * @return number of realizations
 */
public static int getNumRealizations () {
```

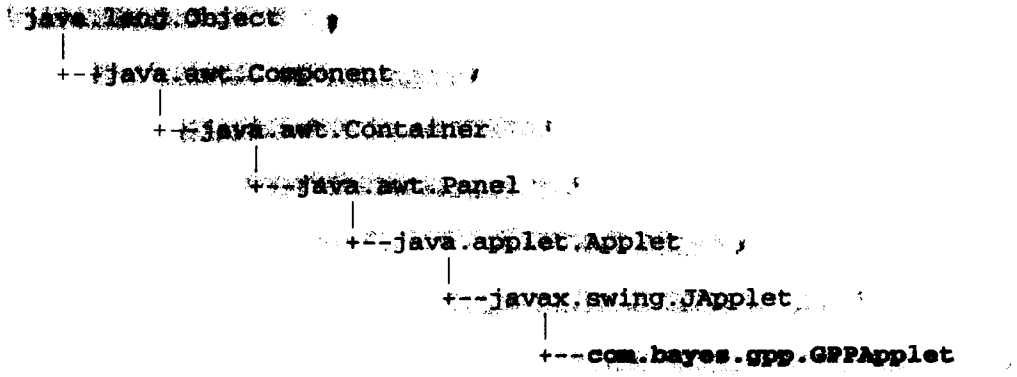
Snapshot created: gpp1_0_20020207-[redacted]521_000.jpg

OK

Example of unique ID of runs.

Class GPP Applet

Class GPPApplet



All Implemented Interfaces:
javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable

---

public class **GPPApplet**
extends javax.swing.JApplet

Program Name:	Graphical Post-Processor for the TPA Code Version 4.0
File Date:	2001-09-03
Release Version:	1.0
Client Name:	U.S. NRC U.S. Nuclear Regulatory Commission NRC Office of Nuclear Material Safety and Safeguards Division of Waste Management
Contract Number:	NRC 02-97-009
NRC Contact:	James Firth (301) 415-6628
CNWRA Contact:	Osvaldo Pensado (210) 522-6084 Center for Nuclear Waste Regulatory Analyses San Antonio, Texas 78238-5166 opensado@swri.edu
Author:	John Emmerling Bayesian Systems, Inc., Gaithersburg, MD

Contact:	Joanne Damours (301) 987-5400 joanne@bayes.com
Documentation:	A Graphical Post-Processor for TPA Code Version 4.0, User's Guide. Version 1.0 Center for Nuclear Waste Regulatory Analyses
NUREG-Series Designator:	N/A

DISCLAIMER

"This computer code/material was prepared as an account of work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Division of Waste Management of the Nuclear Regulatory Commission (NRC), an independent agency of the United States Government. Neither the developer(s) of the code nor any of their sponsors make any warranty, expressed or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represent that its use would not infringe on privately-owned rights."

"In no event unless required by applicable law will the sponsors or those who have written or modified this code, be liable for damages, including any lost profits, lost monies, or other special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by third parties or a failure of the program to operate with other programs), even if you have been advised of the possibility of such damages or for any claim by any other party."

The GPPApplet class is the main frame class for the application, it contains the application menu and also graphically manages the Views, which are a collection of InternalFrames.

This class was implemented by extending JApplet. Any extension of JApplet can be run as a free standing Java program, although it could also be run as an applet provided all the thorny security issues are dealt with.

This is a visual class with many graphical elements. In Java, each such graphical element corresponds to a variable in the code. These variables are generated by JBuilder when used in Design Mode. If necessary for writing the code, the programmer may then rename the variable to reflect the identity of the visual element to which it corresponds. Otherwise, the name is left as generated by JBuilder. When the variable is renamed, the new name given is generally self-explanatory. For all such variables, the javadoc simply identifies them as such. Please use JBuilder's Design Mode as an aid to understanding the use of such variables as part of the implementation of the graphical presentation.

Since:

1.0

See Also:

Serialized Form

Inner Class Summary

(package private) static class	**GPPApplet.FileNameGenerator**
(package private) class	**GPPApplet.RadioListenerTPA**
(package private) class	**GPPApplet.RadioListenerWindow**
(package private) static class	**GPPApplet.SnapThread**

Inner classes inherited from class javax.swing.JApplet

javax.swing.JApplet.AccessibleJApplet

Inner classes inherited from class java.applet.Applet

java.applet.Applet.AccessibleApplet

Inner classes inherited from class java.awt.Panel

java.awt.Panel.AccessibleAWTPanel

Inner classes inherited from class java.awt.Container

java.awt.Container.AccessibleAWTContainer

Inner classes inherited from class java.awt.Component

java.awt.Component.AccessibleAWTComponent, java.awt.Component.AWTTreeLock

Field Summary

private java.awt.BorderLayout	**borderLayout1** Graphical element.
(package private) static net.mirabile.logger.Category	**CFG** For use with Doug's logging package.
(package private) static java.lang.String	**CLS** For use with Doug's logging package.
(package private) static net.mirabile.logger.Category	**EXC** For use with Doug's logging package.
private javax.swing.JFileChooser	**fileChooser** The file chooser dialog used to pick the directory in which the TPA output resides.

private java.awt.FlowLayout	**flowLayout1** Graphical element.
(package private) static net.mirabile.logger.Category	**HST** For use with Doug's logging package.
(package private) boolean	**isStandalone** Indicates that applet is running as a stand alone application.
private javax.swing.JMenu	**jMenu1** Graphical element.
private javax.swing.JMenu	**jMenu2** Graphical element.
private javax.swing.JMenu	**jMenu3** Graphical element.
private javax.swing.JMenu	**jMenu4** Graphical element.
private javax.swing.JMenu	**jMenu5** Graphical element.
private javax.swing.JMenuItem	**jMenuItem1** Graphical element.
private javax.swing.JMenuItem	**jMenuItem2** Graphical element.
private javax.swing.JMenuItem	**jMenuItem3** Graphical element.
private javax.swing.JMenuItem	**jMenuItem4** Graphical element.
private javax.swing.JMenuItem	**jMenuItem5** Graphical element.
(package private) javax.swing.JMenuItem	**jMenuItem6** Graphical element.
private javax.swing.JMenuItem	**jMenuItem7** Graphical element.
private javax.swing.JMenuItem	**jMenuItem8** Graphical element.
private javax.swing.JMenuBar	**mainMenuBar** Graphical element.
private javax.swing.JLabel	**messagesLabel** Graphical element.
private javax.swing.JPanel	**messagesPanel** Graphical element.
private PlotControlFrame	**pcf** Reference to the singleton instance of PlotControlFrame.

(package private) static net.mirabile.logger.Category	**PER** For use with Doug's logging package.
(package private) java.lang.String	**rootDirectory** Not used.
private View	**selectedView** The currently selected View.
private static GPPApplet	**theApplet** Singleton instance of the applet.
private javax.swing.ButtonGroup	**tpaRunButtonGroup** Graphical element.
(package private) static net.mirabile.logger.Category	**TRC** For use with Doug's logging package.
private ViewControlFrame	**vcd** Reference to the singleton instance of ViewControlFrame.
private java.util.Hashtable	**views** The set of all Views displayed by the applet.
private javax.swing.ButtonGroup	**windowButtonGroup** Graphical element.
(package private) javax.swing.JMenu	**windowMenu** Graphical element.
private java.util.Hashtable	**windowMenus** Set of window names (i.e.

Fields inherited from class javax.swing.JApplet
accessibleContext, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.applet.Applet
serialVersionUID, stub

Fields inherited from class java.awt.Panel
base, nameCounter

Fields inherited from class java.awt.Container
component, containerListener, containerSerializedDataVersion, dbg, dispatcher, layoutMgr, listeningBoundsChildren, listeningChildren, maxSize, ncomponents, printing, printingThreads

Fields inherited from class java.awt.Component
actionListenerK, adjustmentListenerK, appContext, background, BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, changeSupport, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, containerListenerK, cursor, dropTarget, enabled, eventMask, focusListener, focusListenerK, font, foreground, graphicsConfig, hasFocus, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, incRate, inputMethodListener, inputMethodListenerK, isInc, isPacked, itemListenerK, keyListener, keyListenerK, LEFT_ALIGNMENT, locale, LOCK, metrics, minSize, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, name, nameExplicitlySet, newEventsOnly, ownedWindowK, parent, peer, peerFont, popups, prefSize, privateKey, RIGHT_ALIGNMENT, textListenerK, TOP_ALIGNMENT, valid, visible, width, windowClosingException, windowListenerK, x, y

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

GPPApplet() Constructor

Method Summary

(package private) static void	()
void	**addView**(java.lang.String viewName, View v) Method to add a View to the main frame and to the Window menu.
void	**destroy**() Applet destroy method
java.lang.String	**getAppletInfo**() A method required to be implemented by an Applet
java.lang.String	**getParameter**(java.lang.String key, java.lang.String def) A method required to be implemented by an Applet.
java.lang.String [][]	**getParameterInfo**() A method required to be implemented by an Applet
static GPPApplet	**getTheApplet**() Accessor method for singleton instance of Applet
void	**init**() Applet init method.
private void	**jbInit**() This is the initialization sequence that is automatically generated and maintained by JBuilder in design mode, it is executed when the class is instantiated.
(package private) void	**jMenuItem1_actionPerformed**(java.awt.event.ActionEvent e) Event handler for "quit" menu item.

(package private) void	**jMenuItem2_actionPerformed**(java.awt.event.ActionEvent e) Event handler for "plot" menu item
(package private) void	**jMenuItem3_actionPerformed**(java.awt.event.ActionEvent e) Event handler for "snap" menu item.
(package private) void	**jMenuItem4_actionPerformed**(java.awt.event.ActionEvent e) Event handler for "view" menu item
(package private) void	**jMenuItem6_actionPerformed**(java.awt.event.ActionEvent e) Event handler for the "dump" menu item
(package private) void	**jMenuItem7_actionPerformed**(java.awt.event.ActionEvent e) Event handler for Help->About menu.
(package private) void	**jMenuItem8_actionPerformed**(java.awt.event.ActionEvent e) Event handler for "Select TPA Run" menu item.
static void	**main**(java.lang.String[] args) Main method, used for unit testing.
static void	**quit**(java.lang.String msg) Method for quitting application
void	**removeView**(java.lang.String viewName) Method to remove a View from the main frame and the Window menu.
void	**start**() Applet start method
void	**stop**() Applet stop method

Methods inherited from class javax.swing.JApplet
addImpl, createRootPane, createRootPaneException, getAccessibleContext, getContentPane, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isRootPaneCheckingEnabled, paramString, processKeyEvent, remove, setContentPane, setGlassPane, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, update

Methods inherited from class java.applet.Applet
getAppletContext, getAudioClip, getAudioClip, getCodeBase, getDocumentBase, getImage, getImage, getLocale, getParameter, isActive, newAudioClip, play, play, resize, resize, setStub, showStatus

Methods inherited from class java.awt.Panel
addNotify, constructComponentName

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, adjustListeningChildren, applyOrientation, checkGD, countComponents, createChildHierarchyEvents, createHierarchyEvents, deliverEvent, dispatchEventImpl, dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt, findComponentAt, getAccessibleAt, getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents_NoClientCode, getComponents, getInsets, getLayout, getListeners, getMaximumSize, getMinimumSize, getMouseEventTarget, getPreferredSize, getWindow, initIDs, insets, invalidate, invalidateTree, isAncestorOf, layout, lightweightPaint, lightweightPrint, list, list, locate, minimumSize, nextFocus, numListening, paint, paintComponents, paintHeavyweightComponents, postProcessKeyEvent, postsOldMouseEvents, preferredSize, preProcessKeyEvent, print, printComponents, printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents, proxyRequestFocus, readObject, remove, removeAll, removeContainerListener, removeNotify, setFocusOwner, setFont, transferFocus, validate, validateTree, writeObject

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, areInputMethodsEnabled, bounds, checkImage, checkImage, checkWindowClosingException, coalesceEvents, contains, contains, createImage, createImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAccessibleIndexInParent, getAccessibleStateSet, getBackground, getBounds, getBounds, getColorModel, getComponentOrientation, getCursor, getDropTarget, getFont_NoClientCode, getFont, getFontMetrics, getForeground, getGraphics, getGraphicsConfiguration, getHeight, getInputContext, getInputMethodRequests, getLocation, getLocation, getLocationOnScreen_NoTreeLock, getLocationOnScreen, getName, getNativeContainer, getParent_NoClientCode, getParent, getPeer, getSize, getSize, getToolkit, getToolkitImpl, getTreeLock, getWidth, getWindowForObject, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isDisplayable, isDoubleBuffered, isEnabled, isEnabledImpl, isFocusTraversable, isLightweight, isOpaque, isRecursivelyVisible, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, resetGC, reshape, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, wait, wait, wait

Field Detail

CFG

```
static net.mirabile.logger.Category CFG
```

For use with Doug's logging package.

EXC

```
static net.mirabile.logger.Category EXC
```

For use with Doug's logging package.

HST

```
static net.mirabile.logger.Category HST
```

For use with Doug's logging package.

PER

```
static net.mirabile.logger.Category PER
```

For use with Doug's logging package.

TRC

```
static net.mirabile.logger.Category TRC
```

For use with Doug's logging package.

CLS

```
static java.lang.String CLS
```

For use with Doug's logging package.

---

isStandalone

```
boolean isStandalone
```

Indicates that applet is running as a stand alone application.

---

rootDirectory

```
java.lang.String rootDirectory
```

Not used.

---

mainMenuBar

```
private javax.swing.JMenuBar mainMenuBar
```

Graphical element.

---

jMenu1

```
private javax.swing.JMenu jMenu1
```

Graphical element.

---

jMenuItem1

```
private javax.swing.JMenuItem jMenuItem1
```

Graphical element.

---

jMenu2

```
private javax.swing.JMenu jMenu2
```

Graphical element.

---

jMenuItem2

```
private javax.swing.JMenuItem jMenuItem2
```

Graphical element.

---

jMenuItem3

```
private javax.swing.JMenuItem jMenuItem3
```

Graphical element.

---

jMenu3

```
private javax.swing.JMenu jMenu3
```

Graphical element.

---

jMenuItem4

```
private javax.swing.JMenuItem jMenuItem4
```

Graphical element.

---

jMenuItem5

```
private javax.swing.JMenuItem jMenuItem5
```

Graphical element.

---

jMenu4

```
private javax.swing.JMenu jMenu4
```

Graphical element.

---

jMenuItem7

```
private javax.swing.JMenuItem jMenuItem7
```

Graphical element.

---

fileChooser

```
private javax.swing.JFileChooser fileChooser
```

The file chooser dialog used to pick the directory in which the TPA output resides.

---

borderLayout1

```
private java.awt.BorderLayout borderLayout1
```

Graphical element.

---

messagesPanel

```
private javax.swing.JPanel messagesPanel
```

Graphical element.

---

flowLayout1

```
private java.awt.FlowLayout flowLayout1
```

Graphical element.

---

messagesLabel

```
private javax.swing.JLabel messagesLabel
```

Graphical element.

---

theApplet

```
private static GPPApplet theApplet
```

Singleton instance of the applet.

---

jMenu5

```
private javax.swing.JMenu jMenu5
```

Graphical element.

---

jMenuItem8

```
private javax.swing.JMenuItem jMenuItem8
```

Graphical element.

---

vcd

```
private ViewControlFrame vcd
```

Reference to the singleton instance of ViewControlFrame.

---

pcf

```
private PlotControlFrame pcf
```

Reference to the singleton instance of PlotControlFrame.

---

tpaRunButtonGroup

```
private javax.swing.ButtonGroup tpaRunButtonGroup
```

Graphical element.

---

windowButtonGroup

```
private javax.swing.ButtonGroup windowButtonGroup
```

Graphical element.

---

views

```
private java.util.Hashtable views
```

The set of all Views displayed by the applet.

---

windowMenus

```
private java.util.Hashtable windowMenus
```

Set of window names (i.e. Views) that can be accessed view the Window menu.

---

selectedView

```
private View selectedView
```

The currently selected View.

---

windowMenu

```
javax.swing.JMenu windowMenu
```

Graphical element.

---

jMenuItem6

```
javax.swing.JMenuItem jMenuItem6
```

Graphical element.

Constructor Detail

GPPApplet

```
public GPPApplet()
```

Constructor

Method Detail

getParameter

```
public java.lang.String getParameter(java.lang.String key,
                                     java.lang.String def)
```

A method required to be implemented by an Applet.

Parameters:

`key` - name of parameter

`def` - default value of parameter

Returns:

parameter value.

---

quit

```
public static void quit(java.lang.String msg)
```

Method for quitting application

Parameters:
`msg` - message to display when quitting

---

init

```
public void init()
```

Applet init method. Calls jbInit.
Overrides:
`init` in class `java.applet.Applet`

---

jbInit

```
private void jbInit()
             throws java.lang.Exception
```

This is the initialization sequence that is automatically generated and maintained by JBuilder in design mode, it is executed when the class is instantiated.
Throws:
`java.lang.Exception` -

---

start

```
public void start()
```

Applet start method
Overrides:
`start` in class `java.applet.Applet`

---

stop

```
public void stop()
```

Applet stop method
Overrides:
`stop` in class `java.applet.Applet`

---

destroy

```
public void destroy()
```

Applet destroy method
Overrides:

destroy in class java.applet.Applet

---

getAppletInfo

```
public java.lang.String getAppletInfo()
```

A method required to be implemented by an Applet

Overrides:

getAppletInfo in class java.applet.Applet

Returns:

Applet information string

---

getParameterInfo

```
public java.lang.String[][] getParameterInfo()
```

A method required to be implemented by an Applet

Overrides:

getParameterInfo in class java.applet.Applet

Returns:

applet parameter values

---

main

```
public static void main(java.lang.String[] args)
```

Main method, used for unit testing.

Parameters:

args -

```
static void ()
```

---

jMenuItem1_actionPerformed

```
void jMenuItem1_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for "quit" menu item.

Parameters:

e - action event

---

jMenuItem2_actionPerformed

```
void jMenuItem2_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for "plot" menu item
Parameters:
`e` - action event

---

jMenuItem4_actionPerformed

```
void jMenuItem4_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for "view" menu item
Parameters:
`e` - action event

---

getTheApplet

```
public static GPPApplet getTheApplet()
```

Accessor method for singleton instance of Applet
Returns:
singleton instance of Applet

---

addView

```
public void addView(java.lang.String viewName,
                    View v)
```

Method to add a View to the main frame and to the Window menu.
Parameters:
`viewName` - name of the the View
`v` - View to be added

---

removeView

```
public void removeView(java.lang.String viewName)
```

Method to remove a View from the main frame and the Window menu.
Parameters:
`viewName` - name of View to be removed

---

jMenuItem8_actionPerformed

```
void jMenuItem8_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for "Select TPA Run" menu item.
Parameters:
e - action event

---

jMenuItem3_actionPerformed

```
void jMenuItem3_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for "snap" menu item. Kicks off an instance of the SnapThread inner class which captures the selected View to a disk file.
Parameters:
e - action event

---

jMenuItem6_actionPerformed

```
void jMenuItem6_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for the "dump" menu item
Parameters:
e - action event

---

jMenuItem7_actionPerformed

```
void jMenuItem7_actionPerformed(java.awt.event.ActionEvent e)
```

Event handler for Help->About menu.
Parameters:
e - action event

~~DataSet.java~~ Constance.java
2/7/2002

Comment lines shown by *

```
/*
 * Title:        1st Graphical Post-Processor Applet<p>
 * Description:  <p>
 * <p>
 * Company:      Bayesian Systems, Inc.<p>
 * @author John Emmerling
 * @version 1.0
 */
package com.bayes.gpp;

/**
 * <table>
 * <tr>
 * <td valign=top>Program Name:</td><td>        Graphical Post-Processor for the
TPA Code Version 4.0</td>
 * </tr>
 * <tr>
 * <td valign=top>File Date:</td><td>           2001-09-03</td>
 * </tr>
 * <tr>
 * <td valign=top>Release Version:</td><td>     1.0</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Client Name:</td><td><table>         <tr><td>U.S.
NRC</td></tr>
 *                      <tr><td>U.S. Nuclear Regulatory Commission</td></tr>
 *                      <tr><td>NRC Office of Nuclear Material Safety and
Safeguards</td></tr>
 *                      <tr><td>Division of Waste
Management</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Contract Number:</td><td>     NRC 02-97-009</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>NRC Contact:</td><td>         James Firth (301) 415-6628</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>CNWRA Contact:</td><td>       <table><tr><td>Osvaldo Pensado
(210) 522-6084</td></tr>
 *                      <tr><td>Center for Nuclear Waste Regulatory
Analyses</td></tr>
 *                      <tr><td>San Antonio, Texas 78238-5166</td></tr>
 *                      <tr><td>opensado@swri.edu</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Author:</td><td>              John Emmerling</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>Bayesian Systems, Inc., Gaithersburg, MD</td>
 * </tr>
 * <tr>
 * <td valign=top>Contact:</td><td> <table><tr><td>Joanne Damours (301) 987-
5400</td></tr>
```

```
 *                                <tr><td>joanne@bayes.com</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Documentation:</td><td>       A Graphical Post-Processor for
TPA Code Version 4.0, User's Guide. Version 1.0</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>                      Center for Nuclear Waste
Regulatory Analyses</td>
 * </tr>
 * <tr>
 *
 *
 * <td valign=top>NUREG-Series Designator:</td><td> N/A</td>
 * </tr>
 * </table>
 *
 *
 *
<center><h2>D I S C L A I M E R</h2></cente
 *
 *
 *     <p>&quot;This computer code/material was prepared as an account of work
 * performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for
 * the Division of Waste Management of the Nuclear Regulatory Commission (NRC),
 * an independent agency of the United States Government. Neither the
 * developer(s) of the code nor any of their sponsors make any warranty,
 * expressed or implied, or assume any legal liability or responsibility for
 * the accuracy, completeness, or usefulness of any information, apparatus,
 * product or process disclosed, or represent that its use would not infringe
 * on privately-owned rights.&quot;</p>
 *
 *     <p>&quot;In no event unless required by applicable law will the sponsors
or
 * those who have written or modified this code, be liable for damages,
 * including any lost profits, lost monies, or other special, incidental or
 * consequential damages arising out of the use or inability to use the program
 * (including but not limited to loss of data or data being rendered inaccurate
 * or losses sustained by third parties or a failure of the program to operate
 * with other programs), even if you have been advised of the possibility of
 * such damages or for any claim by any other party.&quot;</p>
 *
 * <HR>
 *
 * <p>The Constants class was used to define some numbers all of which
 * are now determined at run time.</p>
 *
 * @since 1.0
 * @author John Emmerling
 * @version 1.0
 */
public class Constants {
  /**    */
  public final static int NUM_TIMES = 201;
  /**    */
  public final static int NUM_NUCLIDES =20;
  /**    */
  public final static int NUM_REALIZATIONS = 30;
  /**    */
```

DATASET.JAVA
~~CONSTANTS.JAVA~~
2/7/2002
EXAMPLE

```
/*
 * Title:        1st Graphical Post-Processor Applet<p>
 * Description:  <p>
 * <p>
 * Company:      Bayesian Systems, Inc.<p>
 * @author John Emmerling
 * @version 1.0
 */
package com.bayes.gpp;

import net.mirabile.logger.*; // dbr
import java.io.PrintWriter;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.OutputStreamWriter;
import java.rmi.RemoteException;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.HashSet;
import java.util.Vector;
import visad.BaseUnit;
import visad.CommonUnit;
import visad.FlatField;
import visad.Gridded1DSet;
import visad.FunctionType;
import visad.MathType;
import visad.RealType;
import visad.RealTupleType;
import visad.ScalarType;
import visad.ScaledUnit;
import visad.Set;
import visad.VisADException;
import visad.util.DataUtility;

/**
 * <table>
 * <tr>
 * <td valign=top>Program Name:</td><td>        Graphical Post-Processor for the
TPA Code Version 4.0</td>
 * </tr>
 * <tr>
 * <td valign=top>File Date:</td><td>           2001-09-03</td>
 * </tr>
 * <tr>
 * <td valign=top>Release Version:</td><td>     1.0</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Client Name:</td><td><table>         <tr><td>U.S.
NRC</td></tr>
 *                      <tr><td>U.S. Nuclear Regulatory Commission</td></tr>
 *                      <tr><td>NRC Office of Nuclear Material Safety and
Safeguards</td></tr>
 *                      <tr><td>Division of Waste
Management</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Contract Number:</td><td>     NRC 02-97-009</td>
```

```
 * </tr>
 * <tr>
 *
 * <td valign=top>NRC Contact:</td><td>         James Firth (301) 415-6628</td>
 * </tr>
 * <tr>
 *
 * <td valign=top>CNWRA Contact:</td><td>       <table><tr><td>Osvaldo Pensado
(210) 522-6084</td></tr>
 *                        <tr><td>Center for Nuclear Waste Regulatory
Analyses</td></tr>
 *                        <tr><td>San Antonio, Texas 78238-5166</td></tr>
 *                        <tr><td>opensado@swri.edu</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Author:</td><td>              John Emmerling</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>Bayesian Systems, Inc., Gaithersburg, MD</td>
 * </tr>
 * <tr>
 * <td valign=top>Contact:</td><td> <table><tr><td>Joanne Damours (301) 987-
5400</td></tr>
 *                                <tr><td>joanne@bayes.com</td></tr></table></td>
 * </tr>
 * <tr>
 *
 * <td valign=top>Documentation:</td><td>       A Graphical Post-Processor for
TPA Code Version 4.0, User's Guide. Version 1.0</td>
 * </tr>
 * <tr>
 * <td valign=top> </td><td>                        Center for Nuclear Waste
Regulatory Analyses</td>
 * </tr>
 * <tr>
 *
 *
 * <td valign=top>NUREG-Series Designator:</td><td> N/A</td>
 * </tr>
 * </table>
 *
 *
 *
<center><h2>D I S C L A I M E R</h2></cente
 *
 *
 *     <p>&quot;This computer code/material was prepared as an account of work
 * performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for
 * the Division of Waste Management of the Nuclear Regulatory Commission (NRC),
 * an independent agency of the United States Government. Neither the
 * developer(s) of the code nor any of their sponsors make any warranty,
 * expressed or implied, or assume any legal liability or responsibility for
 * the accuracy, completeness, or usefulness of any information, apparatus,
 * product or process disclosed, or represent that its use would not infringe
 * on privately-owned rights.&quot;</p>
 *
 *     <p>&quot;In no event unless required by applicable law will the sponsors
or
 * those who have written or modified this code, be liable for damages,
```

```
 * including any lost profits, lost monies, or other special, incidental or
 * consequential damages arising out of the use or inability to use the program
 * (including but not limited to loss of data or data being rendered inaccurate
 * or losses sustained by third parties or a failure of the program to operate
 * with other programs), even if you have been advised of the possibility of
 * such damages or for any claim by any other party.&quot;</p>
 *
 * <HR>
 *
 * <p>The DataSet class encapsulates the "model" part of a display that is to be
 * rendered using VisAD.  It collects the VisAD objects that have to do with the
 * data being displayed rather that how the data is to be displayed, for example
 * the data points have been gathered into FlatFields, also there are MathTypes
 * and FunctionTypes contained here.</p>
 *
 * <p>The motivation for creating this class was the hope that it might be
 * possible to make the program sophisticated enough to distinguish between
times
 * when the user merely desired changes of a visual nature e.g. background color
 * versus times when reorganization of the data itself is requested e.g.
breakout
 * by parameter value.  In practice, most requested changes to a plot
specification
 * tend to be of the latter type which results in having to recreate all
DataSets
 * associated with a plot.</p>
 *
 * @since 1.0
 * @author John Emmerling
 * @version 1.0
 */
public class DataSet {
  /** For use with Doug's logging package.
   */
  static Category PER = GPPApplet.PER;  // performance
  /** For use with Doug's logging package.
   */
  static String CLS="DataSet. ";
  // Copied from Plot
  /** <p>A utility copied from TPAData and modified slightly.  Should be part
   * of some utilities package.  Its purpose is to determine unique names
   * for RealTypes created by this DataSet that also conform to a
   * reasonable naming convention.</p>
   *
   * <p>Has special logic for when the name parameter is "year".</p>
   *
   * @param name Name, typically name of the variable being plotted.  Will be
   *     name actually used in most cases.
   * @return Unique name used to identify this RealType.
   */
  public static RealType findOrMakeRType(String name)
  { RealType rt = null;
    name = name.replace(' ','_');
    try
    { rt = RealType.getRealTypeByName(name);
      if (rt == null)
      { if (name.equals("year"))
        { double day_secs = 86400.0, hr_secs = 3600.0, min_secs = 60.0;
          double yr_secs = // 1 greg yr = 365 dys, 5 hrs, 49 min, 12 sec
            365.0 * day_secs + 5.0 * hr_secs + 49.0 * min_secs + 12.0;
```

```
          ScaledUnit time_unit = new ScaledUnit(yr_secs, (BaseUnit)
              CommonUnit.second, "year");
          rt = new RealType("year", time_unit, null);
        }                                   // name == "year"
        else
          rt = new RealType(name);
      }                                     // not already a type of this name.
    } catch (VisADException e) { e.printStackTrace(); }

    return rt;
  }                                         // findOrMakeRType()

  /** The set of all instances of this class, indexed by a unique name
   * actually determined with help from the TPAData class.
   */
  private static Hashtable AllDataSets = new Hashtable();

  /** ScalarType associated with the X coordinate (typically always "year").
   */
  private RealType coord_T;
  /** ScalarType associated with the Y coordinate (typically always "year").
   * Not used for anything.
   */
  private RealType coord_V;

  /** Function type associated with each FlatField (should be same for all
FlatFields)
   */
  private FunctionType curve_ft = null;
  /** FunctionType mapping the curves (realizations or quantiles) to time.
   */
  private FunctionType curves_ft;
  /** RealTupleType used in constructing curve_ft
   */
  private RealTupleType curve_rng_t;
  /** Gridded1DSet associated with (time) domain, used to construct
   * FlatFields.
   */
  private Set curve_dom_set = null;

  /** Not used.
   */
  private Gridded1DSet domain_set_T  = null;

  /** Set of FlatFields that make up this DataSet.  They represent either
   * realizations or quantiles depending on whether or not the data is
   * regularized.
   */
  private FlatField[] realizations = null;

  /** An instance of TPAData that has been rolled up with respect to
   * subarea, nuclide, etc., using elements of tpaDatas.
   */
  private TPAData rolledUp = null;
  /** An instance of TPAData that has been rolled up with respect to
   * subarea, nuclide, etc., using elements of tpaDatas.
   * This instance
   * will subsequently be converted to log values to produce the data
   * member rolledUp, if a log scale is desired.
   */
```

```
private TPAData rawRolledUp = null;

// FArray domain = null;
/** The domain data typically a set of time points at logarithmic
 * intervals.
 */
private PlottableData domain = null;

/** The instances of TPAData passed in to the constructor, from
 * which the FlatFields are constructed.
 */
private TPAData[] tpaDatas;

/** The unique name of this instance of DataSet.
 */
private String name = null;

/** Indicates that this DataSet is a "clone".  See the split method.
 */
private boolean cloneFlag = false;

/** Indicates that the dependent data is in log representation.
 */
private boolean log = false;

/** The minimum and maximum values of the dependent variable across
 * the entire DataSet.
 */
private float[] yminmax = null;

/** The minimum and maximum values of the dependent variable across
 * the entire DataSet, for non-logarithmic values where logarithmic
 * representation is being used.
 */
private float[] yminmax_raw = null;

private float nonzero_min = Float.MAX_VALUE;

// bogus DataSet that has no FlatFields
// kludge for handling scatter plots
/** Kludge for handling scatter plots
 * bogus DataSet that has no FlatFields.
 *
 * @param b
 */
public DataSet(TPAData b) {
  realizations = new FlatField[0];
  tpaDatas = new TPAData[1];
  tpaDatas[0] = b;
}

/** Constructor that creates a DataSet from a single TPAData.
 * Used only for the "scatter plot".
 *
 * @param b
 * @param domain
 */
public DataSet(TPAData b, PlottableData domain)
{ NDC.push(CLS+"ctor1 ");
  ILog.info(PER, "enter " );  // dbr
```

```
    this.domain = domain;
    coord_T = findOrMakeRType("year");
    coord_V = findOrMakeRType(b.getName());
    tpaDatas = new TPAData[1];
    tpaDatas[0] = b;
    try {
      int nRealizations = 0;
      if (!b.isEmpty()) {
        nRealizations = tpaDatas[0].nRows();
      }
      System.out.println("nRealizations: "+nRealizations);
      realizations = new FlatField[nRealizations];
      RealType     curve_dom_rt      = domain.getRealType();
      Set curve_dom_set = new Gridded1DSet(curve_dom_rt, domain.get(),
domain.nRows());
      RealType curves_dom_rt = domain.findOrMakeRType("index");
      RealType curve_rng_rt = ((PlottableData)tpaDatas[0]).getRealType(0);
                // check name, is it "year"?
      // System.out.println("curve_rng_rt: "+curve_rng_rt);
      curve_rng_t = new RealTupleType(curve_rng_rt);
      FunctionType curve_ft = new FunctionType(curve_dom_rt, curve_rng_t );
      curves_ft = new FunctionType( curves_dom_rt, curve_ft);
      // System.out.println("curves_ft: "+curves_ft);
      // System.out.println("Domain RealType: "+domain.getRealType());
      // System.out.println("domain.nRows(): "+domain.nRows());
      // System.out.println("curve_dom_set: "+curve_dom_set);
      for (int q=0; q<nRealizations; q++) {
        // realizations[q] = new FlatField( func_T_V,  domain_set_T /*
Gridded1DSet */ );
        realizations[q] = new FlatField( curve_ft,  curve_dom_set );
        /* EXPLORATORY
        float[][] ff = tpaDatas[0].getCols(q, 0);
        System.out.println("Result of call to tpaDatas[0].getCols("+q+"):");
        for (int ii=0; ii<ff.length; ii++) {
          System.out.print("ii: "+ii+" ");
          for (int jj=0; jj<ff[ii].length; jj++) {
            System.out.print(ff[ii][jj]+" ");
            if (0 == (jj % 20)) {
              System.out.println("");
            }
          }
          System.out.println("");
        }
        // END EXPLORATORY */
        realizations[q].setSamples(tpaDatas[0].getCols(q, 0));
      }
    } catch (VisADException exc) {
      exc.printStackTrace();
    } catch (RemoteException exc) {
      exc.printStackTrace();
    }
    yminmax = getYminmax();
    // FOR THIS TO WORK, TPAData must implement "getName()" which returns some
sort
    // of unique name.  Could be a combo of variable name, row name, col name...
    name = tpaDatas[0].getName();
    AllDataSets.put(name,this);
    ILog.info(PER, "exit  " );  // dbr
    NDC.pop();
  }
```

```
// This method "clones" the DataSet except that the realizations are split
// up between the "parent" and the "clone"
/** This method "splits" the DataSet.  Deprecated.
 *
 * @param rzns set of realizations to include in "clone".
 * @return DataSets resulting from the split
 * @deprecated
 */
public DataSet[] split (HashSet rzns)
{ NDC.push(CLS+"split ");
  ILog.info(PER, "enter ");  // dbr

  // rocket science this ain't
  DataSet clone = new DataSet();
  clone.cloneFlag = false;
  clone.yminmax = this.yminmax;
  clone.coord_T = this.coord_T;
  clone.coord_V = this.coord_V;
  clone.curves_ft = this.curves_ft;
  clone.curve_rng_t = this.curve_rng_t;
  clone.domain_set_T = this.domain_set_T;
  clone.domain = this.domain;
  clone.name = this.name;
  clone.tpaDatas = new TPAData[this.tpaDatas.length];
  for (int i=0; i<this.tpaDatas.length; i++) {
    clone.tpaDatas[i] = this.tpaDatas[i];
  }
  // now for the fun part!
  clone.realizations = new FlatField[rzns.size()];
  FlatField[] newRzns = new FlatField[getSize()-rzns.size()];
  int rznCounter1 = 0;
  int rznCounter2 = 0;
  for (int i=0; i<getSize(); i++) {
    if (rzns.contains(new Integer(i))) {
      clone.realizations[rznCounter1++] = this.realizations[i];
    } else {
      newRzns[rznCounter2++] = this.realizations[i];
    }
  }
  DataSet clone2 = new DataSet();
  clone2.cloneFlag = true;
  clone2.yminmax = this.yminmax;
  clone2.coord_T = this.coord_T;
  clone2.coord_V = this.coord_V;
  clone2.curves_ft = this.curves_ft;
  clone2.curve_rng_t = this.curve_rng_t;
  clone2.domain_set_T = this.domain_set_T;
  clone2.domain = this.domain;
  clone2.name = this.name;
  clone2.tpaDatas = new TPAData[this.tpaDatas.length];
  for (int i=0; i<this.tpaDatas.length; i++) {
    clone2.tpaDatas[i] = this.tpaDatas[i];
  }
  clone2.realizations = newRzns;
  DataSet[] ret = new DataSet[2];
  ret[0] = clone; ret[1] = clone2;
  ILog.info(PER, "exit  " );  // dbr
  return ret;
}
```

```
/** Gets a "slice" of the DataSet containing realizations included in
 * argument.  Deprecated.
 *
 * @param rzns set of realizations to include in "slice"
 * @return DataSet representing "slice"
 * @deprecated
 */
public DataSet getSlice (HashSet rzns) {
  // rocket science this ain't
  DataSet clone = new DataSet();
  clone.cloneFlag = false;
  clone.yminmax = this.yminmax;
  clone.coord_T = this.coord_T;
  clone.coord_V = this.coord_V;
  clone.curves_ft = this.curves_ft;
  clone.curve_rng_t = this.curve_rng_t;
  clone.domain_set_T = this.domain_set_T;
  clone.domain = this.domain;
  clone.name = this.name;
  clone.tpaDatas = new TPAData[this.tpaDatas.length];
  for (int i=0; i<this.tpaDatas.length; i++) {
    clone.tpaDatas[i] = this.tpaDatas[i];
  }
  // now for the fun part!
  clone.realizations = new FlatField[rzns.size()];
  FlatField[] newRzns = new FlatField[getSize()-rzns.size()];
  int rznCounter1 = 0;
  int rznCounter2 = 0;
  for (int i=0; i<getSize(); i++) {
    if (rzns.contains(new Integer(i))) {
      clone.realizations[rznCounter1++] = this.realizations[i];
    }
  }
  return clone;
}

/** Accessor method for curves_ft.
 *
 * @return curves_ft
 */
public FunctionType get_curves_ft () {
  return curves_ft;
}

/*
public RealTupleType get_curve_rng_t () {
  return curve_rng_t;
}
*/

/** Default constructor.
 */
public DataSet() {
}

/** To determine if data is regularized.
 *
 * @param i index of index of tpaDatas in question
 * @return whether data is regularized
```

```
   */
  public boolean isRegularizedData(int i) {
    return tpaDatas[i] instanceof RegularizedTPAData;
  }

  // public DataSet(TPAData[] b, FArray domain) {
  /** Constructor that creates a DataSet from multiple TPADatas.
   * Used in most cases.
   *
   * @param b
   * @param domain
   * @param log
   */
  public DataSet(TPAData[] b, PlottableData domain, boolean log)
  { NDC.push(CLS+"ctor2 ");
    this.log = log;
    ILog.info(PER, "enter " + Integer.toString(b.length) + " ");  // dbr
    // System.out.println("In DataSet CTOR, # of TPAData's: "+b.length);
    this.domain = domain;
    coord_T = findOrMakeRType("year");
    TPAData zerothTPA = b[0];
    coord_V = findOrMakeRType(zerothTPA.getName());
    int nRealizations = 0;
    if (!zerothTPA.isEmpty()) {
      nRealizations = zerothTPA.nRows(); // not necessarily realizations, just
curves
      // System.out.println("DataSet has "+nRealizations+" realizations");
    }
    int len = b.length;
    System.out.println("len: "+len);
    try {
     for (int i=0; i<len; i++) {
        // System.out.println("Rolling up TPAData # "+i);
        if (i==0) {
          rolledUp = b[0].makeDeepCopy();
          // rolledUp = new
TPAData(zerothTPA.getVariable(),0,0,"dog","cat","fish");
          // System.out.println("nRealizations: "+nRealizations);
          realizations = new FlatField[nRealizations];
          RealType     curve_dom_rt      = domain.getRealType();
          curve_dom_set = new Gridded1DSet(curve_dom_rt, domain.get(),
domain.nRows());
          RealType curves_dom_rt = domain.findOrMakeRType("index");
          RealType curve_rng_rt = ((PlottableData)zerothTPA).getRealType(0);
                // check name, is it "year"?
          // System.out.println("curve_rng_rt: "+curve_rng_rt);
          this.curve_rng_t = new RealTupleType(curve_rng_rt);
          curve_ft = new FunctionType(curve_dom_rt, curve_rng_t );
          curves_ft = new FunctionType( curves_dom_rt, curve_ft);
          // System.out.println("curves_ft: "+curves_ft);
          // System.out.println("Domain RealType: "+domain.getRealType());
          // System.out.println("domain.nRows(): "+domain.nRows());
          // System.out.println("curve_dom_set: "+curve_dom_set);
        } else {
          for (int q=0; q<nRealizations; q++) {
            rolledUp.addArrayRowTo(b[i],q,0);
          }
        }
        // System.out.println("i: "+i+", rolledUp nRows: "+rolledUp.nRows()+",
nCols: "+rolledUp.nCols());
```

```
    } // end for
    if (len > 1) {
      for (int q=0; q<nRealizations; q++) {
       rolledUp.divideRowByConstant((float)len,q,0);
      }
    }
    if (log) {
     rawRolledUp = rolledUp.makeDeepCopy();
     convertToLog();
    } else {
     rawRolledUp = rolledUp;
    }
    nRealizations = realizations.length;
    for (int q=0; q<nRealizations; q++) {
     realizations[q] = new FlatField( curve_ft,  curve_dom_set );
     realizations[q].setSamples(rolledUp.getCols(q, 0));
    }
   } catch (VisADException exc) {
     exc.printStackTrace();
   } catch (RemoteException exc) {
     exc.printStackTrace();
   }
   int size = b.length;
   tpaDatas = new TPAData[size];
   for (int i=0; i<size; i++) {
     tpaDatas[i] = b[i];
   }
   yminmax = getYminmax();
   name = zerothTPA.getName();
   AllDataSets.put(name,this);
   ILog.info(PER, "exit "); // dbr
   NDC.pop();
  }

/** Accessor function for domain.
 *
 * @return domain
 */
public PlottableData getDomain() { return domain; }

/** Accessor function for domain_set_T.
 *
 * @return domain_set_T
 */
public Gridded1DSet getDomainSet() { return domain_set_T; }

/** Retrieves the requested FlatField.
 *
 * @param i index of request FlatField
 * @return requested FlatField
 */
public FlatField getRealization(int i) { return realizations[i]; }

/** Accessor method for coord_T
 *
 * @return coord_T
 */
public ScalarType getXcoord() { return coord_T; }

/** Accessor method for coord_V
```

```
 *
 * @return coord_V
 */
public ScalarType getYcoord() { return coord_V; }

/** Determines number of realizations in DataSet.
 *
 * @return number of realizations in DataSet.
 */
public int getSize() { return realizations.length; }

/** Accessor method for cloneFlag
 *
 * @return cloneFlag
 */
public boolean isClone () { return cloneFlag; }

/** Determines scale color for plot.
 *
 * @return scale color
 */
public float[] getScaleColor()
{
  return ((PlottableData)tpaDatas[0]).getScaleColor();
}

/** determines RealType for DataSet.
 *
 * @return RealType for DataSet
 */
public RealType getRealType()
{
  return ((PlottableData)tpaDatas[0]).getRealType();
}

/** RealType for DataSet at a particular "depth", probably not needed.
 *
 * @param d "depth"
 * @return RealType for DataSet
 */
public RealType getRealType(int d) {
  return ((PlottableData)tpaDatas[0]).getRealType(d);
}

// Only used for scatter plot kludge!!
/** Accessor method for tpaDatas.
 *
 * @return tpaDatas
 */
public TPAData[] getTPADatas() {
  return tpaDatas;
}

/** Used to convert rolled up data to log representation.
 */
private void convertToLog () {
  /* TRYING SOMETHING DIFFERENT
  float[] minMax = rolledUp.getMinMax(0);
  double tiny = (double)minMax[1] * 1.0e-06;
  double smallest = (double)minMax[0];
```

```
    // System.out.println("minMax: ("+minMax[0]+","+minMax[1]+"), tiny:
"+tiny+", smallest: "+smallest);
    if (smallest < tiny)
      smallest = tiny;
    double log_smallest = Math.log(smallest)/FArray.LOG10;
    // dbr log_smallest is the value to substitute for the output
    // of taking the log if the input is too small (incl. neg).
    // System.out.println("smallest: "+smallest+", log_smallest:
"+log_smallest);
    float f_smallest=(float)smallest, f_log_smallest=(float)log_smallest;
    if (f_smallest == 0.0f) {
      f_smallest = Float.MIN_VALUE;
    }
    System.out.println("f_smallest: "+f_smallest+", f_log_smallest:
"+f_log_smallest);
    END TRYING SOMETHING DIFFERENT */
    float f_smallest = this.getNonZeroMin();
    float f_log_smallest = (float)(Math.log((double)f_smallest)/FArray.LOG10);
    for (int q=0; q<rolledUp.nRows(); q++)
    { // for (int q=0; q<TPARun.getNumRealizations(); q++) {
      rolledUp.convertRowToLog(q,0,f_smallest, f_log_smallest);
    }
    for (int q=0; q<realizations.length; q++) {
      try {
        realizations[q] = new FlatField( curve_ft,  curve_dom_set );
        realizations[q].setSamples(rolledUp.getCols(q, 0));
      } catch (VisADException exc) {
        exc.printStackTrace();
        System.exit(-1);
      } catch (RemoteException exc) {
        exc.printStackTrace();
        System.exit(-1);
      }
    }

  }

  /** Accessor method for yminmax.  Instantiates same on first call.
   *
   * @return yminmax
   */
  public float[] getYminmax () {
    if (yminmax == null) {
        /*
          System.out.println("In DataSet.getYminmax, size: "+getSize());
          System.out.println("In DataSet.getYminmax, domain.nRows:
"+domain.nRows());
          System.out.println("In DataSet.getYminmax, number of TPADatas:
"+tpaDatas.length);
          */
      yminmax = new float[] {java.lang.Float.MAX_VALUE,-
java.lang.Float.MAX_VALUE};
      if (!isEmpty()) {
        for (int i =0; i< domain.nRows(); i++)
          {
            for (int c = 0; c < getSize(); c++) {
              float v = Float.NaN;
              try {
                v = realizations[c].getFloats(false)[0][i];
              } catch (VisADException exc) {
```

```
                exc.printStackTrace();
              }
              //float v = tpaDatas[0].getVal(i,c,0);
              /*
                if (v == 0.0f) {
                System.out.println("Found a zero!!");
                }
                */
              // System.out.println ("Looking at: "+v+", yminmax[1]:
"+yminmax[1]);
              if (v > yminmax[1]) {
                // System.out.println("Found new max: "+v+" time pt: "+i+", rzn:
"+c);
                yminmax[1] = v;
              }
              if (v < yminmax[0]) {
                // System.out.println("Found new min: "+v+" time pt: "+i+", rzn:
"+c);
                yminmax[0] = v;
              }
            }
          }
      }
    }
    return yminmax;
  }

  /** Accessor method for nonzero_min
   *
   * @return nonzero_min
   */
  public float getNonZeroMin() {
    float[] dog = this.getRawYminmax();
    return nonzero_min;
  }

  /** <p> Accessor method for yminmax_raw.  Instantiates same on first call.
   *
   * <p> Computes smallest non-zero value as a byproduct
   *
   * @return yminmax_raw
   */
  public float[] getRawYminmax () {
    if (yminmax_raw == null) {
      yminmax_raw = new float[] {java.lang.Float.MAX_VALUE,-
java.lang.Float.MAX_VALUE};
      if (!isEmpty()) {
        for (int i =0; i< domain.nRows(); i++)
          {
            for (int c = 0; c < getSize(); c++) {
              float v = Float.NaN;
              v = rawRolledUp.getCols(c, 0)[0][i];
              if (v > yminmax_raw[1]) {
                // System.out.println("Found new max: "+v+" time pt: "+i+", rzn:
"+c);
                yminmax_raw[1] = v;
              }
              if (v < yminmax_raw[0]) {
                // System.out.println("Found new min: "+v+" time pt: "+i+", rzn:
"+c);
```

```
              yminmax_raw[0] = v;
            }
            if (v > 0 && v < nonzero_min) {
              nonzero_min = v;
            }
          }
        }
        if (nonzero_min == Float.MAX_VALUE) {
          nonzero_min = Float.MIN_VALUE;
          realizations = new FlatField[0];
        } else {
          float logval = (float)(Math.log((double)nonzero_min)/FArray.LOG10);
          nonzero_min = (float)Math.pow(10.0,logval-1.0);
        }
      }
    }
    return yminmax_raw;
  }

  /** Convenience function to determine whether empty
   *
   * @return whether empty
   */
  public boolean isEmpty() {
    return getSize() == 0;
  }

  /** Used to produce "ASCII snapshot" of data for user.
   *
   * @param fileName Name of file in which to save snapshot.
   */
  public void dump(String fileName) {
    try {
      PrintWriter out
        = new PrintWriter(new OutputStreamWriter(new
FileOutputStream(fileName,true)));
      PlottableRawData prd = tpaDatas[0].getDomain(log);
      float[][] domainValues = prd.getCol(0);
      out.print("time");
      for (int i=0; i<domainValues[0].length; i++) {
        out.print(","+domainValues[0][i]);
      }
      out.println();
      for (int i=0; i<realizations.length; i++) {
        float[][] values = realizations[i].getFloats();
        out.print(i);
        for (int j=0; j<values[0].length; j++) {
          out.print(","+values[0][j]);
        }
        out.println();
      }
      out.flush();
      out.close();
    } catch (FileNotFoundException exc) {
      exc.printStackTrace();
    } catch (VisADException exc) {
      exc.printStackTrace();
    }
  }
```

```
    /** Gets an instance of this class given the unique instance name.
     *
     * @param name instance name
     * @return DataSet instance
     */
    public static DataSet getInstance(String name) {
        if (!AllDataSets.containsKey(name)) {
          return (DataSet)null;
        } else {
          return (DataSet)AllDataSets.get(name);
        }
    }

    /** Obtains list of names of all instances of DataSet.
     *
     * @return list of names of all instances of DataSet.
     */
    public static String[] getAllNames() {
      int len = AllDataSets.size();
      Enumeration enum = AllDataSets.keys();
      String[] ret = new String[len];
      int i=0;
      while (enum.hasMoreElements()) {
        ret[i++] = (String)enum.nextElement();
      }
      return ret;
    }

    /** Main method, used for unit testing.
     *
     * @param args
     */
    public static void main(String[] args) {
      /*
      DataSet dataSet1 = new DataSet(
          new TPAData("Waste_Package_Temperature",4)
        );
      */
    }
}
```

Software Development Plan

Software Development Plan for
Graphical Post-Processor (GPP) for Total-System Performance Assessment (TPA)

August 7, 2000

This software development plan (SDP) describes the approach to be followed in implementing the Graphical Post-Processor (GPP) for the Total-System Performance Assessment (TPA) application. The implementation will meet the requirements outlined in the Software Requirements Description (SRD) for the GPP.

1.0 SCOPE

The scope is described in detail in the GPP SRD. The work will be performed by Bayesian Systems, Inc. The GPP will be a stand-alone application to be used in conjunction with data generated with the TPA code. Selected TPA output data will be statistically analyzed and the results will interactively be displayed with the help of JAVA applications.

2.0 BASELINE ITEMS

Products to be delivered include: (i) graphical post processor executable, in principle compatible with SUN Solaris and Windows NT operating systems, but only tested in the latter operating system, (ii) source code with development environment, (iii) CVS database for code maintenance, (iv) documentation. Project developed items, as well as items obtained from other sources will be included. The GPP is due by September 6, 2000.

3.0 PROJECT MANAGEMENT

Software development project tasks, schedules, staff and provisions and strategies for reducing associated risk are discussed in this section.

3.1 Work Breakdown Structure

Tasks	Name	Hours	Completion Date
Task 1: Requirements Definition	*Damours and Reingold*		
Determine Design Objectives		160	
Determine Design Approach		60	
Define Plotting functional requirements, including user's plotting controls		20	
Definition of off-the-shelf component evaluation criteria		16	
Evaluation of off-the-shelf plotting components		40	

Trip to San Antonio		32	
Completion Date			*June 30, 2000*
Task 2: Develop Application	*Reingold and Emmerling*		
Implement display windows, Main Window, Plot Window, and View Window		40	
Implement plot into display window		40	
Encode software generated plots into jpeg format for saving as an image file		8	
Implement derivation of percentile curves from raw data		40	
Implement saving of plotted data values in ascii text files		20	
Display the influence of parameters and variables on other variables		40	
Implementation of TPA data file interface		80	
Completion of User's Guide		40	
Completion Date			*August 7, 2000*
Task 3: Testing	*Damours, Reingold and Emmerling*		
Acceptance testing and documentation of testing		80	
Change request tracking		16	
Completion Date			*August 21, 2000*
Task 4: Delivery	*Reingold and Damours*		
CNWRA review of acceptance testing		40	
Prepare media, installation guide, and installation test set		20	
Installation (to be performed by CNWRA staff)			
Completion Date			*Sept. 6, 2000*

3.2 Project schedule and milestones

The main milestone for Tasks 1, completed by June 30, 2000, is the elaboration of the Software Requirements Document (SRD). Task 2, finished by August 7, is aimed at the completion of the core GPP software. By August 21, after completion of Task 3, CNWRA staff will be provided with draft software and documentation to review the acceptance testing. A final version of the code will be delivered by September 6.

3.3 Staffing

Staff from Bayesian Systems contributing to the development of the GPP is listed as follows:

Project Manager – Stan Kaplan
Project Management and Data Analysis – Joanne Damours
System Design and Software Development – Doug Reingold
Software Development – John P. Emmerling, Jr.

3.4 Risk Management

A primary risk associated with this project is accuracy of identification of requirements and their clear communication among the participants and CNWRA staff. The Software Requirements Document will be used to avoid misunderstanding. In addition, continuous communication between the development group and CNWRA staff will reduce the risk of misunderstandings.

A second risk is not deploying the software by September 6, 2000. To insure deployment, Doug Reingold will supervise design and John Emmerling will lead the development and integration effort.

4.0 DEVELOPMENT PROCEDURES

This section describes plans for developing the GPP.

4.1 Environment and Resources

All code development will be done on MS Windows compatible PCs running MS Windows 9x or NT. JAVA will be used as coding language. JBuilder will be used as development tool. VisAD and Colt libraries will be used. Additional information on these libraries is provided as follows.

VisAD is a graphics package from Space Science and Engineering Center, University of Wisconsin. Additional information on VisAD is available at

VisaD Home Page <http://www.ssec.wisc.edu/~billh/visad.html

VisAD has the following two indirect dependencies:

Displays may be generated with either Java2D (included in Java 2) or Java3D. Java3D is freely available from Sun's Java3D web site at

http://java.sun.com/products/java-media/3D/index.html

Java3D requires OpenGL or DirectX display drivers. OpenGL can be freely downloaded for Sparc Solaris and for Windows 95 from the OpenGL web site at

http://www.opengl.org/

OpenGL is a standard display driver part of the Windows NT, 98, 2000, operating systems.

The Colt Distribution Open Source Libraries for High Performance Scientific and Technical Computing in Java have been produced by the High Energy Physics Department at CERN. More information on these libraries, with download links, is available at

http://tilde-hoschek.home.cern.ch/~hoschek/colt/index.htmhttp://tilde-hoschek.home.cern.ch/~hoschek/colt/index.htm

4.2 Software Development Lifecycle

The development lifecycle for the GPP includes five phases: analysis, development, acceptance testing, preliminary release, and final delivery. The analysis phase is aimed at producing a Software Requirements Description (SRD) document. Task 1 in Section 3.1 is part of this phase. Task 2 in Section 3.1 is aimed at addressing the development phase. The core of the GPP software will be produced after completion of this task. The acceptance testing phase will be addressed in Task 3. The acceptance testing will be documented in scientific notebooks or an acceptance test report. The preliminary release phase is also addressed by Task 3. Finally, the final delivery phase is addressed by Task 4. The CNWRA staff will review the acceptance testing and may issue software change requests as result of such revision. This phase also includes assignment of a version number, completion of a verification report commenting on the results of the review of the acceptance testing documents, and release of a software summary form and software release notice. CNWRA staff will perform all these latter tasks. The GPP will be recommended for software validation after tested by staff.

4.3 Coding conventions

Coding will be done in the JAVA 2 standard. Any Java compiler handling the JAVA 2 standard and above will be capable to compile the produced source code.

4.4 Acceptance Testing and Analysis (TOP-18 5.5.7)

The results of the acceptance testing will be appropriately recorded in scientific notebooks and software change reports (SCR).

5.0 CONFIGURATION MANAGEMENT PLAN

The official version of the working code will be placed under control of the Concurrent Version System (CVS) configuration management software. A copy of the final GPP code will be provided to the CNWRA for configuration control, as well as a copy of the CVS database for code maintenance purposes.

5.1 Tools

CVS will be used to perform software configuration management.

5.2 Configuration Identification.

Version 1.0 will be assigned to the GPP.

5.3 Configuration Procedures

The software will be maintained using CVS. The software will reside on the machine called "JOHNE" located at 4927 Clearwater Drive, Ellicott City, MD and in directory e:\gpp. All check-in and check-out activities will be managed by John Emmerling. After initial delivery, the standard SCR form and scientific notebooks will be used to document all changes to the controlled source code. Change comments will be stored in CVS. Files will be transferred from CVS to the distribution media via a CD-ROM burner. Backups will be performed on a daily basis by zipping the CVS database and uploading it to the Bayesian FTP site.

6.0 REFERENCES

None.

7.0 APPENDICES

None.

APPROVED:

Gordon Wittmeyer
Signature of Element Manager

8/17/2000
Date

SOFTWARE REQUIREMENTS DESCRIPTION FOR A GRAPHICAL POST-PROCESSOR (GPP) FOR THE TOTAL-SYSTEM PERFORMANCE ASSESSMENT (TPA) VERSION 4.0 CODE

Prepared for

Nuclear Regulatory Commission
Contract NRC–02–97–009

Prepared by

Stan Kaplan
Douglas Reingold
Joanne Damours

Bayesian Systems, Inc.
Gaithersburg, MD

for

Center for Nuclear Waste Regulatory Analyses
San Antonio, TX

July 2000

Gordon W. Wittmeyer
Performance Assessment

7/14/2000
Date

CONTENTS

FIGURES

1 INTRODUCTION

The purpose of the graphical post-processor (GPP) is to enable the user to develop, from the output of previously run Total-system Performance Assessment (TPA) code cases, graphical displays that will aid in the understanding and communication of the

- Repository performance
- Physical phenomena in the repository
- Effects of these phenomena on the performance
- Effects on performance of various possible modifications to repository design
- Degree of confidence/certainty about each of the above, based on the available evidence

Figures 1-1,1-2, and 1-3 illustrate the GPP output. Figure 1-1 shows the behavior of the key variables throughout the lifetime of the repository. See the glossary at the end of this document to clarify the meaning of relevant terms such as "variable." Taken as a whole, this figure displays the performance of the repository, and it has been, therefore, titled "Repository Performance Summary Diagram (RPSD)." The curves of figure 1-1 represent data aggregated from individual subareas, nuclides, or both.

Recognizing that there is uncertainty about this performance, stemming from uncertainty in modeling the physical phenomena and the future climate conditions, the software will quantify and communicate this uncertainty by displaying a set of RPSDs corresponding to different probability percentiles. For example, the user of the GPP could ask for a set of three RPSDs. One could show a median performance on each of the displayed variables. The second RPSD could show the 90^{th} percentile performance for each of the displayed variables, and the third could show the 10^{th} percentile performance. Note that showing, for example, the 90th percentile performance for multiple variables on the same page does not necessarily imply that such high performance in one variable is associated with high performance in another variable. The grouping of variables and percentiles on a graph is at the user's discretion.

This package of three diagrams would then quantify the state of knowledge of the repository's performance. Similarly, to explore a design change or a different assumption of future rainfall/climate conditions, one will use the GPP to prepare a new set of three (or more as needed) RPSDs to characterize the new performance. For example, figure 1-2 shows the infiltration before and after the construction of a subsurface diversion. The subsequent curves show the effect on the downstream variables before and after diversion.

The GPP will also assist in understanding various cause-effect relationships. For example, a correlation between the mean average temperature increase (MATI) at glacial maximum and the fraction of waste packages failed could be graphically depicted by a representation such as figure 1-3. Figure 1-3 displays a hypothetical positive correlation between the MATI variable and the fraction of waste packages failed.

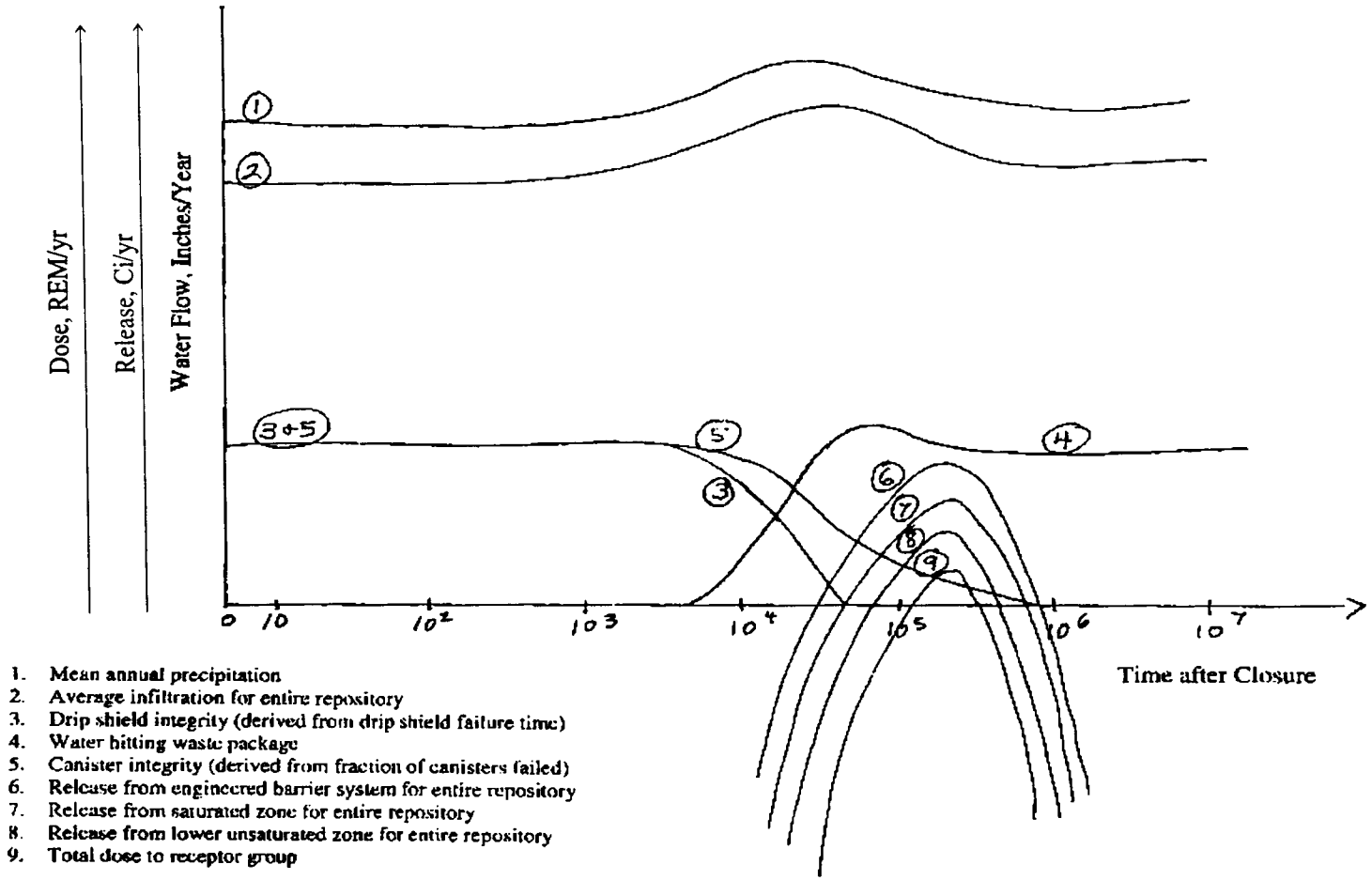
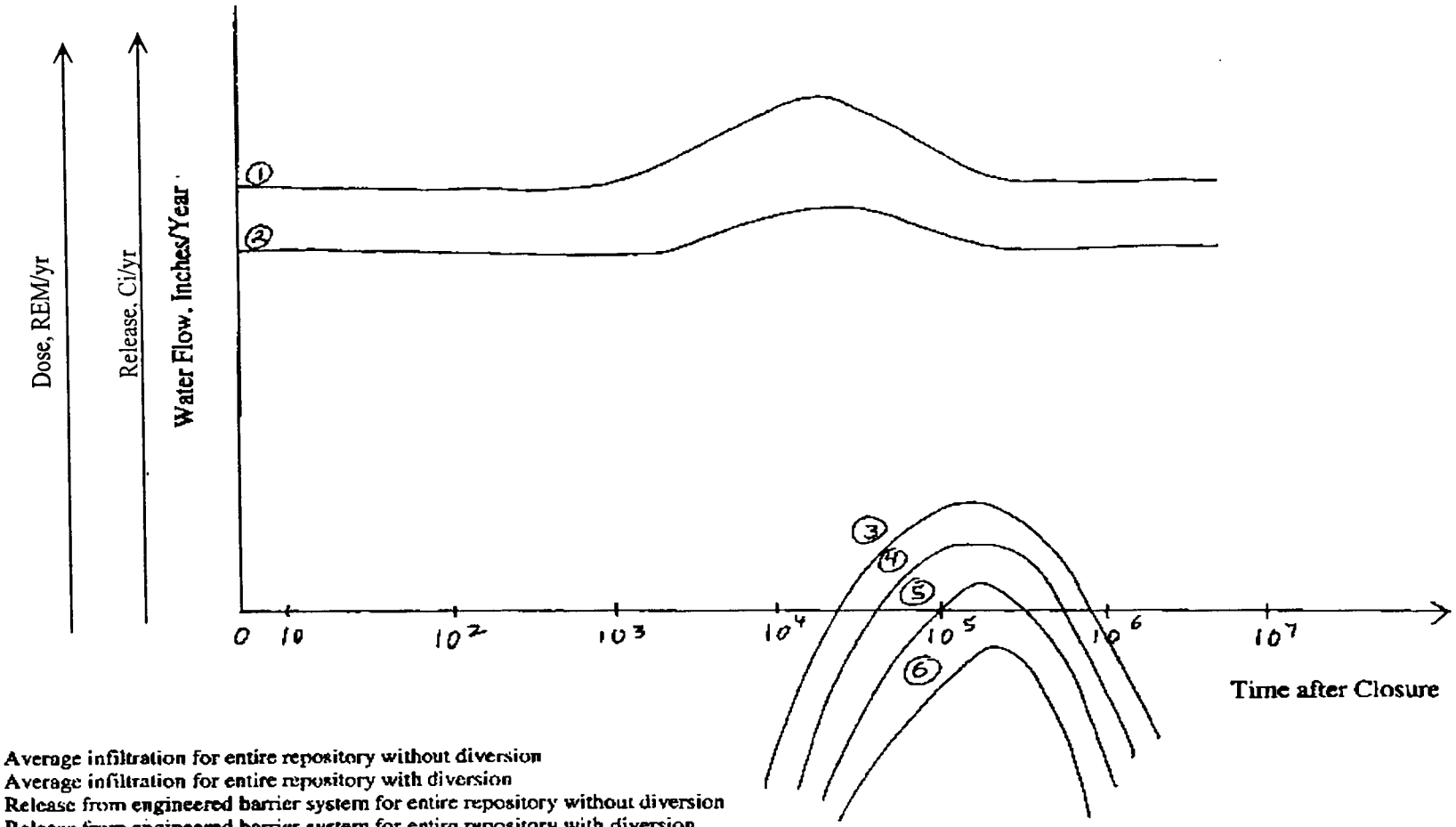


Figure 1-1. Repository performance summary diagram



1. Average infiltration for entire repository without diversion
2. Average infiltration for entire repository with diversion
3. Release from engineered barrier system for entire repository without diversion
4. Release from engineered barrier system for entire repository with diversion
5. Total dose to receptor group without diversion
6. Total dose to receptor group with diversion

Figure 1-2. Repository performance summary diagram: Effects of diversion strategy on water flow

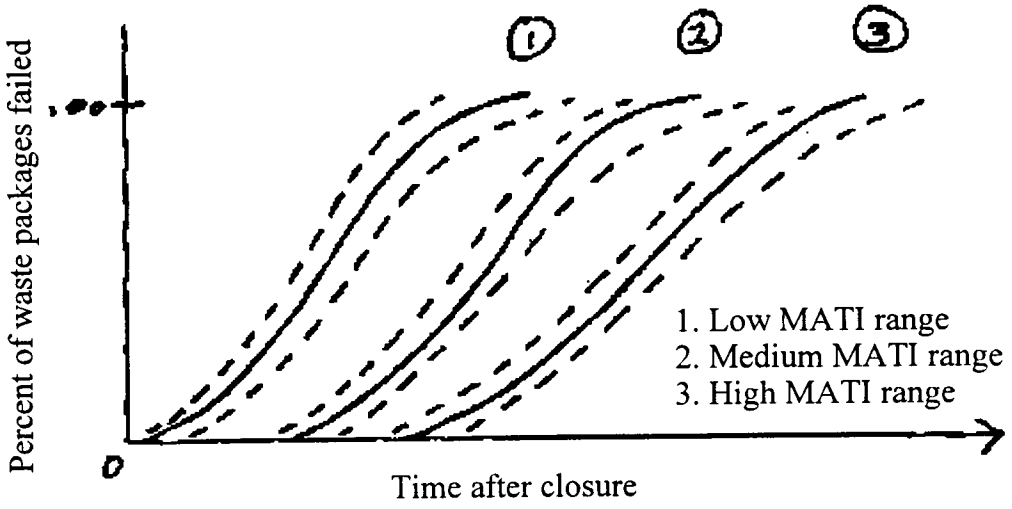


Figure 1-3. Effect of the mean average temperature increase (MATI) at glacial maximum on the percent of waste packages failed. Dotted lines represent uncertainty ranges.

2 SOFTWARE FUNCTION

The software will read TPA output files and graphically depict, for a selected repository configuration (i.e., a TPA code run), the time dependence on any of the following TPA variables

- Mean annual temperature
- Mean annual precipitation
- Infiltration per subarea
- Average infiltration per repository
- Average reflux per repository
- Average diversion per repository
- Fraction of canisters failed
- Drip shield failure time
- Water hitting waste package
- Repository temperature
- Waste package temperature
- Waste package relative humidity
- Release from engineered barrier system per subarea
- Release from engineered barrier system per repository
- Release from lower unsaturated zone per subarea
- Release from lower unsaturated zone per repository
- Release from saturated zone per subarea
- Release from saturated zone per repository
- Current biosphere dose conversion factors
- Pluvial biosphere dose conversion factors

- Total dose to the receptor group
- Peak dose for compliance period

The software will be able to plot for any set of realizations and any of the mentioned variables, either (i) the bundle of trajectories showing for each realization the value of the variable during time, (ii) a single curve summarizing the bundle by representing the median (50th percentile) value of the variable for each time, or (iii) a family of curves containing not only the median curve described previously, but also the curve for any other percentile. The TPA code allows many input parameters to be expressed as a probability distributions that vary over some ranges. The family of curves portrays not only the median trajectory for the variable, but also the combined effect of such variation in input parameters. Such a family of curves enables visualization of the confidence intervals associated with any curve.

The software will graphically depict the degree to which any given TPA input parameter or output variable affects a TPA output variable by

- Enabling the user to select for any TPA output variable, a parameter or variable whose influence the user wants to explore

- Partitioning any set of realizations into subsets based on the range of values for the parameter or variable, and plotting for each subset either the median curve or the family of percentile curves

- As a special case of the previous bulleted item, for any TPA input parameter specified as a distribution, enabling the user to restrict the range of values for that parameter. Such a restriction results in pruning the set of realizations from which the curves are calculated and, therefore, enables the user to visualize the degree to which TPA output may be controlled by any given TPA input parameter.

The capabilities described in the previous paragraph allow the user to ask a wide variety of "what-if" questions concerning how the repository will respond during various interventions or design options. For example, one family may represent the realizations with a high waste package temperature, while another may represent the realizations with a low chloride concentration.

The software will be able to account for the values of any variable, calculated by the TPA code on a per subarea or per nuclide basis.

3 COMPUTATIONAL APPROACH

Standard object oriented design will be employed.

Standard Java coding conventions will be employed.

Custom code will be minimized.

The software will enable the user to

- Control the windows in which plots are displayed

- Control for any plot the appearance of the following visual components
  - Background color
  - Box
  - Legend
  - Title

- Control for any axes of a plot
  - What is the range and kind of scale (linear or logarithmic) used
  - How the axis is labeled
  - What color is used or what variable is mapped to the color
  - What range of values to include
  - Which field is mapped to that axis

- "Snap" or capture, from any plot currently open, an image file that may be stored in "JPEG" or "PNG" format

- Write values for any data being plotted to a disk file in a tabular format

- Select parameters or variables whose influence on the displayed variable the user wants to explore

- Perform side-by-side visual comparisons of different plots

4 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

The TPA GPP is the response to the Advisory Committee for Nuclear Waste (ACNW) suggestion to develop a way to visualize and understand the performance of the repository and the effect of various assumptions and design changes on that performance. Figures 1-1 and 1-2 are examples of the type of graphical output that the GPP will produce.

Figure 1-1 graphically summarizes the overall performance of the repository and the physical phenomena in it. This figure recognizes that the movement of water is the key phenomenon (without which there would be no release of radioactivity). Thus, curve 1 presents the surface rainfall as a function of time. Curve 2 shows infiltration (which could be different from curve 1 if surface drainage were provided). Curve 3 indicates canister integrity and curve 4, drip shield integrity. Curve 5 shows water hitting the waste package. Curve 6 states the release from the engineered barrier system. Curve 7 presents the release from the lower unsaturated zones and curve 8, the release from the saturated zone. Curve 9 indicates the total dose. Ultimately the canister degrades leading to transport of radioactivity to the unsaturated and saturated zones, the biosphere, and the target group.

Figure 1-2 reflects how the performance of the repository would change if a subsurface diversion path were provided. These and similar figures could give understanding of the elements determining repository performance and suggest how that performance could be improved by various interventions or design options. Such a repository change could be simulated by the TPA code by a change in the mean annual infiltration.

GLOSSARY

Throughout this document *variable* represents one of the quantities that a user will want to plot and that is often a function of time.

Parameter refers to the values (specified in file tpa.inp) that affect the variables in TPA realizations.

A *partition* for set S is a collection of disjoint subsets of S such that every element of S is in exactly one of the subsets.

A *trajectory* refers to the projected time history for one variable. This is a curve or scalar function of time.