Documentation for the Domain Editor of the FRAMEwork System (FRAMES)

PNWD-3508

With Contributions From

B. L. Hoopes
M. A. Pelton
K. J. Castleton
G. Whelan
G. M. Gelston

Documentation for the Domain Editor of the FRAMEwork System (FRAMES)

LEGAL NOTICE

This report was prepared by Battelle Memorial Institute (Battelle) as an account of sponsored research activities. Neither Client nor Battelle nor any person acting on behalf of either:

MAKES ANY WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, process, or composition disclosed in this report may not infringe privately owned rights; or

Assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, process, or composition disclosed in this report.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Battelle. The views and opinions of authors expressed herein do not necessarily state or reflect those of Battelle.

Documentation for the Domain Editor of the FRAMEwork System (FRAMES)

Title Page
Legal Notice
Table of Contents
Introduction
Requirements
Design
Quality Assurance and Testing

Documentation for the Domain Editor of the FRAMEwork System (FRAMES)

Title Page
Legal Notice
Table of Contents
Introduction
Requirements
Design
Quality Assurance and Testing

Introduction

This documentation provides information on a component of Version 2.x of the FRAMEwork System (FRAMES), which is a software platform that allows for the linking of various modules into complete assessment systems (Whelan et al. 1997. PNNL-11748). Documentation includes requirements, design and specifications or formulations, and quality assurance and testing.

Portions of this documentation may have been previously issued in reports from the Pacific Northwest National Laboratory (PNNL), operated by Battelle for the U.S. Department of Energy. All PNNL reports are issued a tracking number. Numbers on the title page of this documentation indicate these previous reports.

This documentation can be used by software engineers and testers to ensure that each component functions properly. The information can also be used by analysts and managers to better understand the component's use within FRAMES.

Title Page
Legal Notice
Table of Contents
Introduction
Requirements
Design
Quality Assurance and Testing

Requirements for the FRAMES Domain Editor

The FRAMEwork System (FRAMES) is managed through five editors, each with its own documentation:

- Conversion Editor, which manages measures and units used by dictionaries and datasets to automatically convert between different mediums
- Dictionary Editor, which creates and edits dictionaries used by FRAMES to categorize data
- Module Editor, which creates and manages the modules that manipulate data by specifying dictionaries
- Domain Editor, which organizes a palette of modules
- Simulation Editor, which runs the analysis.

This section summarizes requirements for the Domain Editor. Those interested in setting up a risk scenario and analyzing data should refer to the start up documentation for additional information. Those interested in importing a module into FRAMES should refer to the importation documentation for additional information.

The Domain Editor provides the capability to customize a FRAMES 2.0 session. In FRAMES 2.0, domains are nothing more than an organization of modules on a palette. Domains are designed with the understanding that certain modules can be grouped together, either in a group or subgroup within a domain's class. In previous versions of FRAMES, only one domain existed, and groups were static. With FRAMES 2.0, multiple domains can be created with any combination of groups and subgroups. The Domain Editor provides the capability of setting up the palette by managing the main components on the palette. The main components are described below:

- **Domains** - an organization of modules on a palette. A domain contains four base classes and an icon that will be displayed in FRAMES 2.0. The icon can be modified, but none of the classes can be added or deleted from the domain.
- **Classes** - the type of component. There are four classes present in FRAMES 2.0: Model, System, Database, and Viewer. The classes depict how a module is represented in FRAMES.
- **Groups** - one of two locations where a module can be added to a domain. A group may contain multiple subgroups, modules, and an icon to be displayed in FRAMES 2.0.
- **Subgroups** - the other location where a module can be added to a domain. A subgroup contains an icon that will be displayed in FRAMES 2.0 and is capable of containing multiple modules. Any module that does not declare an icon in the Module Editor will inherit the subgroup's icon.

The Domain Editor allows the user to manage the main components of the domain. As requirements, the Domain Editor will allow the user to perform the following tasks:

1. Add a domain
2. Edit a domain
3. Delete a domain
4. Set a domain icon
5. Add a class
6. Edit a class
7. Delete a class
8. Set a class icon
9. Add a group
10. Edit a group
11. Delete a group
12. Set a group icon
13. Add a subgroup
14. Edit a subgroup
15. Delete a subgroup
16. Set a subgroup icon.

Title Page
Legal Notice
Table of Contents
Introduction
Requirements
Design
Quality Assurance and Testing

Design of the FRAMES Domain Editor

FRAMES 2.0 allows the user to conduct a risk analysis using modules categorized by classes (models, viewers, databases, and systems). Domains organize these modules in a particular way (a palette) for a particular analysis. Within a domain, a module can be categorized under a group or subgroup within a class. The domain icon can be modified, but none of the classes can be added or deleted from the domain. There can be multiple instances of a module in a domain.

In previous versions of FRAMES, only one domain existed, and groups were static. With FRAMES 2.0, multiple domains can be created with any combination of groups and subgroups. The Domain Editor allows the user to set up a palette by adding, deleting, and changing icons. The following sections describe the design that allows users to manage the palette through domains, groups, and subgroups.

Managing Domains

The FRAMES analysis starts with a domain. Each analysis requires at least one domain. To create a domain, the user must add a domain in the Domain Editor by selecting the root (top) node called "Domains." A list of domains will appear in the panel on the right-hand side. The user clicks the "Add Domain" button. When the input box appears, the user enters a name for the new domain. On confirmation, the Domain Editor adds a new domain along with the four base classes.

Once a domain has been added, the user can manipulate it in several ways, such as deleting it, editing its icon to distinguish it from another domain, or editing the icon belonging to a class to show a more narrow analysis. To delete a domain, the user selects the root (top) node and highlights a domain from the available list. The user then clicks on the "Delete Domain" button. On confirmation, the Domain Editor removes the domain and classes from the available list.

To edit a domain's icon, the user selects a domain and clicks on the "Edit Domain Icon" button. On the Open File form, the user locates and opens an icon. The Domain Editor modifies the icon to the one selected.

To edit a class's icon, the user selects a class under a domain and clicks on the "Edit Class Icon" button. On the Open File form, the user locates and opens an icon. The Domain Editor modifies the icon to the one selected.

Managing Groups

A group may contain multiple subgroups, modules, and an icon to be displayed in FRAMES 2.0. Groups and subgroups are the only locations where a module can be added to a domain. Groups can be managed by adding a group, deleting a group, editing a group's icon, adding a module to a group, and deleting a module from a group.

To add a group to the domain, the user selects a class under the domain on the user interface and clicks the "Add Group" button. In the input box, the user then enters a name for the new group. On confirmation, the Domain Editor adds a new group.

To delete a group from the domain, the user selects a class under the domain, then highlights the group from the available list. The user clicks on the "Delete Group" button. On confirmation, the Domain Editor removes the group from the list.

To edit a group's icon, the user selects a group and click's on the "Edit Group Icon" button. In the Open File form, the user then locates and opens an icon. The Domain Editor modifies the icon.

To add a module to a group, the user selects a group and clicks on the "Add Module" button. The user then locates a module from the list and clicks on the "Add Module" button again. The Domain Editor adds the module to the available list. Only modules belonging to a specific class are available to be added. A module's class and the domain's class in which the group or subgroup resides must match in order to add the module.

Note that when a module is added to a group or subgroup, the user may not modify the icon to be displayed on the FRAMES' palette. If the module has an icon associated with it, that icon will be displayed. However, if a module does not specify an icon, then an icon will automatically be determined from the group or subgroup to which the module belongs.

To delete a module from a group, the user selects a group and then highlights a module from the available list. On confirmation, the Domain Editor will remove the module from the list.

Managing Subgroups

A subgroup may also contain multiple modules and an icon that will be displayed in FRAMES 2.0. Any module that does not declare an icon when it was added to the system will inherit the subgroup's icon. Subgroups may be managed by adding a subgroup, deleting a subgroup, editing a subgroup's icon, adding a module to a subgroup, and deleting a module from a subgroup.

To add a subgroup to a domain, the user selects a group in the user interface and clicks on the "Add Subgroup" button. On the input box, the user then enters a name for the new subgroup. On confirmation, the Domain Editor adds the new subgroup.

To delete a subgroup, the user selects a group and highlights a subgroup from the available list. The user then clicks on the "Delete Subgroup" button. On confirmation, the Domain Editor deletes the subgroup.

To edit a subgroup's icon, the user selects the subgroup and clicks on the "Edit Subgroup Icon" button. On the Open File form, the user locates and opens an icon. The Domain Editor modifies the subgroup's icon.

To add a module to a subgroup, the user selects a subgroup and clicks on the "Add Module" button. On the list of modules, the user then locates a module and clicks on the "Add Module" button again. Only modules belonging to a specific class are available to be added. A module's class and the class in which the subgroup resides must match in order to add the module. The Domain Editor then adds the module to the subgroup.

Note that when a module is added to a subgroup, the user may not modify the icon to be displayed on the FRAMES' palette. If the module has an icon associated with it, that icon will be displayed. However, if a module does not specify an icon, then an icon will automatically be determined from the group or subgroup to which the module belongs.

To delete a module from a subgroup, the user selects a subgroup and highlights a module from the available list. The user then clicks on the "Delete Module" button. On confirmation, the Domain Editor removes the module from the list.

Title Page
Legal Notice
Table of Contents
Introduction
Requirements
Design
Quality Assurance and Testing

Quality Assurance and Testing of the FRAMES Domain Editor

The FRAMES Domain Editor was developed under a quality assurance (QA) program that looked at the software life cycle: requirements analysis, design, programming, modification, testing, and implementation. Part of the QA program involves testing each component to ensure that it satisfies its requirements. The requirements section of this documentation provides a list of requirements for the FRAMES Domain Editor. A test plan was developed with test cases that addressed these requirements. An overview of quality assurance and testing for FRAMES can be found in the system documentation.

For additional information on the QA program, including testing, see the following documents:

- *An Approach to Ensuring Quality in Environmental Software* (Gelston et al. 1998. PNNL-11880)
- *Test Plan for the FRAMES Domain Editor*
- Status Report for the FRAMES Domain Editor
- Files used for testing the FRAMES Domain Editor.