

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

PNWD-3504

With Contributions From

B. L. Hoopes
M. A. Pelton
K. J. Castleton

[Title Page](#)
[Legal Notice](#)
[Table of Contents](#)
[Introduction](#)
[Requirements](#)
[Design](#)
[Quality Assurance and Testing](#)



[Home](#) | [Security and Privacy](#) | [Contact Us](#)

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

[Title Page](#)
[Legal Notice](#)
[Table of Contents](#)
[Introduction](#)
[Requirements](#)
[Design](#)
[Quality Assurance and Testing](#)

[Title Page](#)

[Legal Notice](#)

[Introduction](#)

[Requirements](#)

[Design](#)

Formatting Dictionary Files

Privileges

Data Type

Scalar

Minimum and Maximum

Measure and Unit

Stochastic

Preposition

Index

Manipulating Dictionaries

Publishing Dictionaries Using the Linkage Server

[Quality Assurance and Testing](#)

Quality Assurance Plan

[Test Plan](#)

[Testing Status](#)

[Files Needed for Testing \(Test Bed\)](#)



[Home](#) | [Security and Privacy](#) | [Contact Us](#)

Documentation for the Dictionary 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.



[Home](#) | [Security and Privacy](#) | [Contact Us](#)

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

[Title Page](#)
[Legal Notice](#)
[Table of Contents](#)
[Introduction](#)
[Requirements](#)
[Design](#)
[Quality Assurance and Testing](#)

Requirements of the FRAMES Dictionary Editor

Dictionaries in FRAMES 2.0 describe the layout of how information is stored in a dataset. Each dictionary has a name, description of itself, privilege information, and variables. All dictionaries in FRAMES 2.0 must have a unique name. Each dictionary is stored in its own .DIC file.

Dictionaries are also used by modules to define the limitations and capability of a model's inputs and outputs. Modules create connection schemes of dictionaries to describe the type of information a module expects to consume and produce (see [Module Editor documentation](#) for more information).

The Dictionary Editor allows the user to manage dictionaries within FRAMES 2.0. The Dictionary Editor has the following requirements:

1. Provide a dictionary (.DIC) file interface for creating, editing, and deleting dictionary files.
2. Require that all .DIC files have a name and privilege information and provide an option to provide description information.
3. Register .DIC files that are created outside of the FRAMES upon opening them in FRAMES.
4. Automatically attach the .DIC file extension to new .DIC files created in FRAMES.
5. Require a user name and password before allowing a user to edit system DIC files.
6. Require a user with security clearance to specify a privilege setting for each DIC file created using FRAMES. The four privilege levels are 0 - system developer module inputs, 1 - system developer boundary conditions, 2 - module developer module inputs, and 3 - module developer boundary conditions.
7. Ensure that each DIC file has a unique name.
8. Allow the user to add, edit, and delete variables in a DIC file.
9. Ensure that each variable in a DIC file has a unique name.
10. Allow the user to specify variable properties (i.e., data type, minimum and maximum, measure, units, stochastic, and preposition).
11. Allow the user to specify variables that use other parameters as indices.
12. Allow the user to specify that a variable is a primary key.
13. Allow the user to specify if a variable is a scalar or a vector.



[Home](#) | [Security and Privacy](#) | [Contact Us](#)

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

[Title Page](#)
[Legal Notice](#)
[Table of Contents](#)
[Introduction](#)
[Requirements](#)
[Design](#)
[Quality Assurance and Testing](#)

Design of the FRAMES Dictionary Editor

All information about a dictionary is stored in an ASCII file with a .DIC file extension. When the Dictionary Editor changes information about a dictionary, it is actually manipulating the dictionary's .DIC file. This section describes the format for dictionary files, using privileges and variables, manipulating dictionary files, and publishing dictionary files to a server. In-depth information about dictionaries can be found in a separate [dictionary document](#).

Formatting Dictionary Files

Each dictionary is stored in an ASCII file. The file can reside anywhere on the system. Dictionaries can be created outside the Dictionary Editor using any kind of text editor. However, dictionaries created outside the Dictionary Editor must register their dictionary with the editor (see Opening a Dictionary below) to be recognized by the system.

When creating a dictionary within the FRAMES 2.0 Dictionary Editor, a .DIC file extension is automatically assigned to the dictionary file. FRAMES 2.0 does not require that a dictionary contain a .DIC file extension; however, the Dictionary Editor only recognizes .DIC file extensions as dictionaries.

The file format of a dictionary is as follows:

```
Line 1 - Description of dictionary fields
Line 2 - [Variable Count],[name],[description],[privilege]
Line 3 - Description of variable fields
Line 4 on -
[Name],[Description],[Dimension],[Data Type],[Scalar],
[Minimum],[Maximum],[Measure],[Unit],[Stochastic],
[Preposition],[Index 1],[Index 2], Y [Index 6]
```

While many of these items will be familiar to most module or system developers, the privilege, data type, scalar, minimum and maximum, measure and unit, stochastic, preposition, and index items are used in specific manners within FRAMES, as described in the subsections below.

In general, all information in FRAMES 2.0 is stored in datasets, and datasets must use variables to store their information correctly. A variable defines what type of data is going to be added, how the data are retrieved (in a list or a single value), the type of units (if any), and other variables needed to select a particular value with a set of information.

To add a variable to a dictionary, the user opens the Dictionary Editor and selects a dictionary. Clicking on the "New Variable" button brings up an input box. The user enters the name of the new variable. After confirmation, the new variable will appear on the list of associated variables.

To delete a variable from a dictionary, the user opens the Dictionary Editor and selects a dictionary. A list of all the variables belonging to the selected dictionary appears on the right. The user highlights a variable from the list and clicks on the "Delete Variable" button. After confirmation, the Dictionary Editor removes the variable from the list.

Privileges

Each dictionary contains a privilege property field that describes the dictionary's scope. A dictionary's scope is determined by who created the dictionary (system developer or module developer) and by the context of its use within FRAMES. System developer dictionaries are created and used by FRAMES 2.0 and should only be modified by individuals with appropriate access privileges. For example, a module developer will not have access privileges to change any information in a system developer dictionary but will have access privileges to view and reference system developer dictionaries.

The privilege field also describes the context of dictionary use. A dictionary may be interpreted as a model input dictionary or a boundary condition dictionary. A model input dictionary informs FRAMES that a specific module will use the dictionary, whereas a boundary condition dictionary informs FRAMES that the dictionary will be used as global information for all modules.

To make changes to system developer dictionaries and system modules, the user must first log in as a system developer. The log in form is accessed via the File menu.

Data Type

Data can be stored in four different ways:

- logical - [true|false]
- float - [0.01549|87| -32.223]
- integer - [1|546|-97]
- string - ["Name" | "123"].

Scalar

The scalar field describes a true or false value used to identify retrieval method. Data can be retrieved as single values or a list of values. If the scalar value is false, then a single value will be returned. If the scalar value is true, then a list of values will be returned. If the scalar value is set to true, the last indexing variable must be self-referencing. FRAMES will not provide any checks for this condition. It is up to the user to know and maintain this protocol.

Minimum and Maximum

When defining a variable, restraints can be placed on the data to indicate acceptable values. These restraints are accomplished through the minimum and maximum fields. Minimum and maximum values mean different things for different data types:

- logical values - no effect
- float values - [min > = float > = max]
- integer values - [min > = integer > = max]
- string values - [min > = length of string > = max].

In addition, as shown in the table below, there are special cases for some minimum and maximum values.

Special Cases of Minimum and Maximum Values

Minimum	Maximum	Meaning
0	*blank	0 > = value = MAX
*blank	0	MIN > = value > 0
0	0	MIN > value > = MAX

* Cannot be represented within the Framework Development Environment.
User must use text editor to make changes.

Measure and Unit

Data within a variable can specify the type of measure/unit. This ability allows FRAMES 2.0 to automatically convert between mediums without addition input from the user. All data associated with a variable within a dictionary must use the same unit of measure. For example, it is not possible for a single variable in a dictionary to treat some values as **seconds** and others as **nanoseconds**. See the [documentation for the Conversion Editor](#) for additional information.

Stochastic

Stochastic is a true or false value that signifies whether a variable's data can be modified. If the stochastic flag is set to true, the variable's data cannot not be modified.

Preposition

The preposition of a variable is used to describe, in English, how a variable is used in communication with other variables. Typical prepositions for a variable include the "for," "at," or "is" prepositions.

Index

Variables have the capability to reference other variables to set and retrieve data. Currently in FRAMES 2.0, a variable has a limit of referencing up to six other variables. Variables can only reference other variables with a dimension less than two (meaning that a variable can only reference against a list or singular value). The order in which indexing variables are defined is used to identify how to store and retrieve data. Each indexing variable acts as a filter over the data in question. If more than one indexing variable exists, the data can be thought of as going through multiple filters until only one solution remains.

When adding an indexing variable, the starting variable depends on other data to add or retrieve data from its location in a dataset. To add an indexing variable using the Dictionary Editor, the user selects a variable and clicks the "Add Index" button, which brings up a form. The user then selects the variable to reference and clicks the "Add Index Variable" button.

To delete an indexing variable using the Dictionary Editor, the user selects a variable and then an associated indexing variable and clicks on the "Del Index" button. On confirmation, the Dictionary Editor will remove the indexing variable.

Because the order of indexing variables is important to adding and retrieving data, the Dictionary Editor includes the ability to "promote" and "demote" an indexing variable up or down in the list. To promote an indexing variable, the user selects the variable and associated indexing variable and clicks the "Up" button. To demote an indexing variable, the user selects the variable and associated indexing variable and clicks the "Down" button.

Manipulating Dictionaries

Dictionaries themselves can also be manipulated from the Dictionary Editor, including adding a dictionary and deleting a dictionary. Dictionaries can be added in two ways: by creating an entirely new dictionary and by opening a dictionary that already exists but has not been added to the list. In both cases, FRAMES uses system files to store information about the dictionary.

To add a new dictionary, the user selects the appropriate node under system developer or model developer (for boundary condition or model input dictionaries, respectively) to select privilege, then clicks on the "New Dictionary" button. On the Open File form, the user enters a name that does not already exist. On confirmation, the Dictionary Editor will prompt the user to create the file. On confirmation, the Dictionary Editor creates a new file, which should appear in both the list and in the tree beneath the current selected node.

To open an existing dictionary, the user selects the appropriate node under system developer or model developer (for boundary condition or model input dictionaries, respectively) to select privilege, then clicks on the "Open Dictionary" button. On the Open File form, the user selects a dictionary that has not already been added to the list and opens the file. The Dictionary Editor adds the new dictionary to the list and tree beneath the current selected node.

To delete a dictionary, the user selects the appropriate node under system developer or model developer (for boundary condition or model input dictionaries, respectively) to select privilege, highlights the dictionary of interest, then clicks on the "Delete Dictionary" button. On confirmation, the Dictionary Editor deletes the selected dictionary unless that dictionary has been referenced by another dictionary or is used by any module its connection scheme.

Publishing Dictionaries Using the Linkage Server

The Linkage Server is an online database for publishing and retrieving FRAMES modules and dictionaries . FRAMES automatically checks for the latest version when a user attempts to publish a dictionary to the Linkage Server. If the Linkage Server's version is newer, a message box prompts the user for confirmation to overwrite the online dictionary.

To publish a dictionary to the Linkage Server from the Dictionary Editor, the user selects a dictionary from the list of available dictionaries and clicks the "Publish Online" button. To download a dictionary from the Linkage Server, the user selects the Linkage Server from the Tools menu, bringing up the Linkage Server Updates form. Modules appear at the left, with associated dictionaries beneath them. Icons associated with a dictionary or module in the Linkage Server Updates form define the relationship between files on the computer where FRAMES is residing and files on the Linkage Server database. The user selects a dictionary from the Linkage Server Updates form and clicks the "Update Dictionary" button. The Linkage Server downloads the dictionary into FRAMES.



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

- [Title Page](#)
- [Legal Notice](#)
- [Table of Contents](#)
- [Introduction](#)
- [Requirements](#)
- [Design](#)
- [Quality Assurance and Testing](#)

Quality Assurance and Testing of the FRAMES Dictionary Editor

The FRAMES Dictionary 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 Dictionary Editor. An overview of quality assurance and testing for FRAMES can be found in the [system documentation](#).

A test plan was developed with test cases that addressed these requirements. The table below shows how these requirements were addressed in testing.

Testing Matrix for the FRAMES Dictionary Editor

Requirement	Test Case	
	1	2
---	1	2
1	Y	-
2	Y	-
3	Y	-
4	-	Y
5	-	Y
6	-	Y
7	-	Y
8	-	Y
9	-	Y
10	-	Y
11	-	Y
12	-	Y
13	-	Y

An Approach to Ensuring Quality in Environmental Software ([Gelston et al. 1998](#), PNNL-11880)

- [Test Plan for the Framework Development Environment](#)
- [Status Report for the Framework Development Environment](#)
- [Test bed for the Framework Development Environment](#).

