# TREK

# METADATA

# USER GUIDE


**November 2024**

TABLE OF CONTENTS

PARAGRAPH PAGE

# FIGURES

FIGURE PAGE

# 1 Welcome

The Telescience Resource Kit (TReK) is a suite of software applications and libraries that can be used to monitor and control assets in space or on the ground.

The TReK Metadata application provides the capability to create and manage telemetry and command metadata. This includes support for creating databases and metadata files and translating between different types of metadata formats.

The topics in this user guide require an understanding of the topics covered in the TReK Concepts document. Please be sure you have read the TReK Concepts document before reading this user guide.

## 1.1 Getting Started

Start with the Introduction which provides an application overview. Next, try the Quick Start Guides for "How Tos" for common functions. For help with details, reference the Details section. See the FAQ and Troubleshooting section for helpful hints and solutions to the common "gotchas".

# 2 Technical Support

If you are having trouble installing the TReK software or using any of the TReK software, please contact us for technical assistance:

E-Mail: trek.help@nasa.gov

Messages sent to this address are automatically forwarded to the TReK team.

The HOSC Help Desk (256-544-5066) can provide assistance as needed and is available 24x7.

# 3 Introduction

The TReK Metadata application can be used to create and manage telemetry and command metadata files including creating databases and translating between metadata formats. For a detailed introduction to TReK metadata, reference the TReK Concepts document.

Packets describe data sent from one system to another system. Commands and telemetry applications send and receive packets. Command data is packets that tell another system to do something. Telemetry data is packets that supply information about the system sending the packet.

Within a packet, parameters are individual data values. Parameters may contain information about the state of the sending system or actions to be taken by the receiving

system. Parameter values are either placed in outgoing packets or pulled from incoming data packets. In TReK, the terms “build” is placing parameters in a packet and “extract” is pulling parameters from a packet.

Parameter collections are groups of related parameters. Parameter collections are the basic building blocks of packets. Figure 1 has four rows illustrating how parameters are grouped into collections within a packet. The first row shows a packet as a single entity. The second row shows that the packet is composed of parameter collections and another packet. The third row shows the packet will break down into a series of parameter collections. The final row shows all of the parameters. Each row is a different view of the same data.

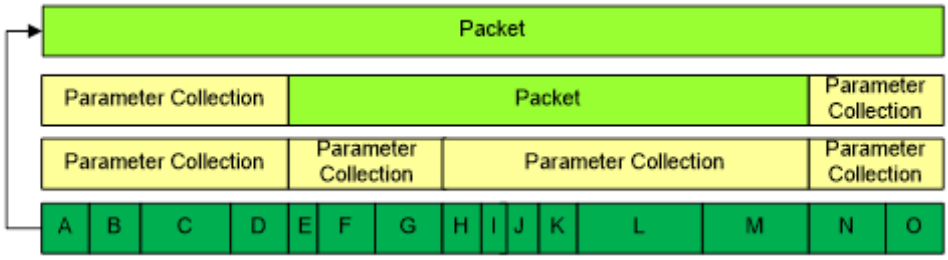


**Figure 1 Packets, Parameter Collections, and Parameters**

Packet collections are given one of three zones: header, data, or trailer. A valid packet must have at least one packet collection in one of the zones. Figure 2 breaks down an illustration of a packet with all three zones where the data zone contains another packet defining two of the zones.


**Figure 2 A Packet and Its Zones**

The first line in the figure above shows a packet that has all three zones defined. The second level shows that the header and trailer zone are composed of parameter collections which contain one or more parameters. The data zone is composed of another packet which only has the header and data zones defined. The third line shows that the packet in the data zone of the top level packet is composed of two parameter collections and that all of the data in a packet will eventually break down into a series of parameters.

The Metadata application provides the capability to compose, from the bottom up, a parameter collection into the zones of either a telemetry packet or a command packet. It provides all the support needed to define the details for a Packet, Parameter Collection, and Parameters like those shown in Figure 1 and Figure 2. Once defined these packet definitions can then be used with various TReK applications.

Since there are many ways to format metadata, TReK supports a variety of formats as described below.

C Programming Language Header File
Many software developers represent a packet definition using a C structure defined in a C header file. TReK can import the packet definition by reading the C structure defined in a C header file. See Section 6.31 for details.

Common Data Exchange Format (CDEF)
CDEF is an International Space Station (ISS) program format that contains metadata described using XML. It is used to exchange telemetry and command information between data systems. See Section 6.30 for details.

TReK Database
A TReK database is a SQLite database that can contain metadata information for one or more packets.

TReK Metadata File
This is a TReK generated file that contains metadata described using XML (e.g., a Packet definition, a Parameter Collection definition).

The Metadata application can read, write, and translate between these different types of formats.

The Metadata application also provides the capability to import metadata information obtained from the HOSC including metadata defined in Partial Database Download Files and Ground Support Equipment (GSE) packet definitions.

# 4 Overview of the User Interface

## 4.1 Main Window

The main window contains several areas as shown in Figure 3. The Toolbar contains multiple buttons that reconfigure the main window to focus on a specific metadata element such as a Parameter Collection, a Database, a Limit Alarm, a Packet, etc.

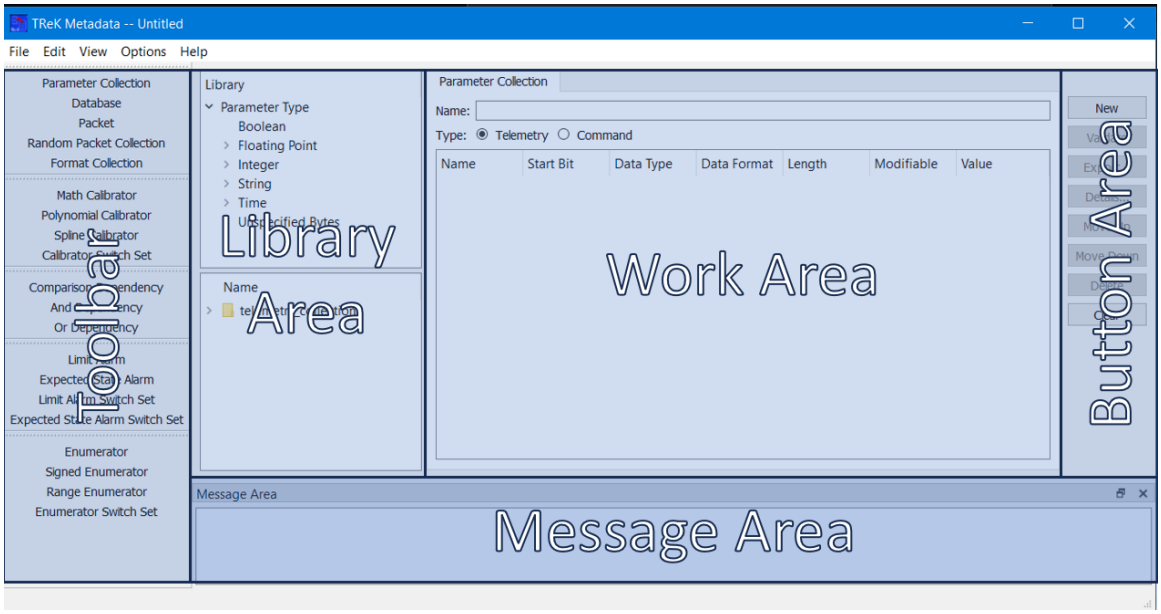


**Figure 3 Main Window**

Toolbar

The toolbar at the left provides quick access to reconfigure the main window to work on a specific metadata element.

Library Area

The Library Area is context sensitive and will be reconfigured to support the specific area you are working on. For example, if you are creating a Parameter Collection, the Library Area will be configured to show a list of all the types of parameters that you can add to a parameter collection. If you are working on a Database, the Library will be configured to show you a list of databases you can work with.

Work Area

The Work Area is context sensitive and will be reconfigured to support the specific type of element you are working on. If you are working on a Parameter Collection, the Work Area will display the Parameter Collection content. If you are working on a Database, the Work Area will display the list of packets in the Database you have selected. If you are working on a Limit Alarm, the Work Area will display the content of the Limit Alarm. If you are working on a Packet, the Work Area will display the contents of the packet. The Work Area provides the capability to create or modify the contents of a specific element.

Button Area

The Button Area is context sensitive and will be reconfigured to display functions available for the specific area you are working on. For example, for a Parameter Collection, you would have functions such as creating a New Parameter Collection, Validating a Parameter Collection, and Exporting the Parameter Collection (saving it in a file in a specified format. There are also functions available to modify the order of

parameters in the collection by moving them up and down in the list and deleting a parameter.

Message Area
The Message Area displays important status and error messages. The message area can be cleared using the View menu. The Message Area is a dock window that you can float or dock. To float a dock window, use your left mouse button to click and hold the title area while dragging the window to another area of the screen. To dock, use the title bar to drag the dock window over the main window and drop.

### 4.2 Menus

The Metadata application menus are: File, View, Options, and Help. Each of these menus is described in more detail below.

File Menu
The File menu provides the capability to manage configurations and exit the application.

View Menu
The View menu provides the capability to clear the main window message area and show and hide different areas in the main window.

Options Menu
The Options menu provides access to the Messages dialog and other special functions.

Help Menu
The Help menu provides access to on-line help and application version information.

## 5 Quick Start Guides

This section provides "How Tos" for common functions.

### 5.1 How to Create a Parameter Collection

The following steps describe the minimum necessary to create a Parameter Collection. For additional information and details please reference section 6.1.

1. Push the Parameter Collection button to configure the Work Area to create a Parameter Collection.
2. Enter a name for the Parameter Collection.
3. Select the Telemetry or Command radio button to specify whether the Parameter Collection will be a telemetry collection or a command collection.
4. Add one or more parameters to the Parameter Collection by dragging a Parameter Type from the Library Area and dropping it in the Parameter Collection List.
5. Enter the minimum information necessary to define a Parameter: Name, Start Bit, Data Type, and Length.

6. Validate the Parameter Collection.
7. Export the Parameter Collection to save a copy in a file.

## 5.2 How to Create a Database

The following steps describe the minimum necessary to create a TReK Database. For additional information and details please reference section 6.2.

1. Push the Database button to configure the Work Area to work with a TReK Database.
2. Select the Telemetry or Command radio button to specify whether the Database will be a telemetry database or a command database.
3. Push the New button and enter the name for the new Database. Once you push the OK button to close the New Database dialog, the Database will be opened and ready for editing in the Work Area.

## 5.3 How to View the List of Packets in a Database

The following are the minimum necessary steps to view the list of Packets in a TReK Database. For additional information and please reference section 6.2.

1. Push the Database button to configure the Work Area to work with a TReK Database.
2. Select the Telemetry or Command radio button to indicate the type of Database. The Library will reconfigure to list the selected type of databases.
3. Select a Database item in the Library and drag it to the Name field and drop it. The packets in the Database will be displayed in the list.

Note: The Library displays telemetry databases that reside in the TReK workspace telemetry_database folder and command databases that reside in the TReK workspace command_database folder.

## 5.4 How to Export a Packet from a Database

This section describes how to export a packet definition from a TReK Database. For additional information and details please reference section 6.2.

1. Push the Database button to configure the Work Area to work with a TReK Database.
2. Select the Telemetry or Command radio button to indicate the type of Database. The Library will reconfigure to list the selected type of databases.
3. Select a Database item in the Library and drag it to the Name field and drop it. The packets in the Database will be displayed in the list.
4. Select a Packet in the list and push the Export button.
5. In the Export dialog, select a Format, enter the required information, and push OK to export the packet definition.

## 5.5 How to Add a Packet to a Database

This section describes how to add a packet definition to a TReK Database using the TReK Database ASCII file format. For additional information and details please reference section 6.2.

1. Push the Database button to configure the Work Area to work with a TReK Database.
2. Select the Telemetry or Command radio button to indicate the type of Database. The Library will reconfigure to list the selected type of databases.
3. Select a Database item in the Library and drag it to the Name field and drop it. Any packets in the Database will be displayed in the list.
4. Push the Import button to add a packet to the database.
5. In the Import dialog select the "Packet Definition in ASCII File Format" option and browse to select the file(s) describing the packet and then push the Import button. If the packet was successfully imported, it will appear in the list or an error message will be displayed.

## 5.6 How to Create a Limit Alarm

The following steps describe the minimum necessary to create a Limit Alarm. For additional information and details please reference section 6.9.

1. Push the Limit Alarm button to configure the Work Area to create a Limit Alarm.
2. Enter a name for the Limit Alarm.
3. Enter a type for the Limit Alarm.
4. Enter the information for one or more levels in the Limit Alarm.
5. Validate the Limit Alarm.
6. Export the Limit Alarm to save a copy in a file.

## 5.7 How to Create a Packet

The following steps describe the minimum necessary to create a Packet. For additional information and details please reference section 6.3.

1. Push the Packet button to configure the Work Area to create a Packet.
2. Enter a name for the Packet.
3. Select the Telemetry or Command radio button to specify whether the Packet will be a telemetry packet or a command packet.
4. Drag and drop Parameter Collections from the library onto the packet zone headers above the packet definition area. For example, start the packet definition by using a predefined packet template. Then add or update the packet zones to create a custom packet. [Hint: You can drag items such as collections to the individual drop zones such as the Header drop zone, Data drop zone, or Trailer drop zone, or you can right click on a Header, Data, or Trailer in the packet definition to get a pop-up menu to Add, Replace, or Delete the contents of a zone.
5. Validate the Packet.
6. Export the Packet to save a copy in a file.

## 5.8 How to Import Data from HOSC Partial Database Download Files into a TReK Database

The following steps describe the minimum necessary to import data from HOSC Partial Database Download files into a TReK database. The data must be imported into a new TReK database. For additional information and details please reference section 6.3.

1. Go to the Options menu and select 'Convert EHS Database Files to TReK Database'.
2. In the Convert EHS Database Files to TReK Database dialog select Telemetry or Command based on the type of data to import.
3. In the TReK Database Information area, enter the name and directory for the new TReK database that will be created to hold the information from the EHS database files.
4. In the EHS Partial Database File Information, enter the information that identifies where the EHS partial Database files can be found.

   Note: Filenames for EHS Partial Database files with telemetry information contain a prefix number. This number should be entered in the Prefix field.

5. Push the Convert button to execute the import. Messages related to the convert process will be displayed in the Results area. The process can take a while if the EHS partial database files contain a large amount of data. You can continue to work on other tasks in the application while the convert is in work.

## 5.9 How to Import a GSE Packet Definition into a TReK Database

This section describes how to import a HOSC Ground Support Equipment (GSE) packet definition into a TReK Database. Information about GSE packets can be found in the Payload to Generic User Interface Definition Document (PGUIDD).

1. Push the Database button to configure the Work Area to work with a TReK Database.
2. Select the Telemetry radio button. The Library will reconfigure to list telemetry databases.
3. You can add the GSE packet definition to a new Telemetry Database or to an existing Telemetry Database listed in the library.
   - To create a new database
     a. Push the New Button.
     b. Enter a name for the new database and push OK.
   - To select an existing Telemetry Database, select the item in the Library and drag it to the Name field and drop it. Any packets in the Database will be displayed in the list.
4. Push the Import button to add a packet to the database.
5. In the Import dialog select the "EHS GSE Packet File in ASCII File Format" option and browse to select the file(s) describing the GSE packet and then push the Import

button. If the packet was successfully imported, it will appear in the list, or an error message will be displayed.

### 5.10 How to add an EHS PDSS Payload Pass-Thru Packet to a TReK Database

This section describes how to add an EHS PDSS pass-thru packet to a TReK Database. If your packet is defined as a pass-thru packet, you can use the TReK API to retrieve the header parameters or the data zone portion of the packet, but not any of the individual parameters in the data zone of the packet. Once you have added a pass-thru packet to a TReK Database, you can use capabilities on the Metadata application's Database tab to export the packet definition to a TReK metadata file.

1. Go to the Options menu and select 'Add EHS PDSS Payload Pass-Thru Packet to TReK Database'.
2. In the 'Add EHS PDSS Payload Pass-Thru Packet to TReK Database' dialog, enter the database information. You can use an existing database or create a new database.
3. In the Packet Information area, enter the APID for your packet.
4. Push the Apply button. The results will be displayed in the Results area.

## 6 Details

This section covers various application details.

### 6.1 Parameter Collection

The Parameter Collection configuration provides the capability to work with a Parameter Collection. The Main Window with the Parameter Collection configuration is shown in Figure 4.

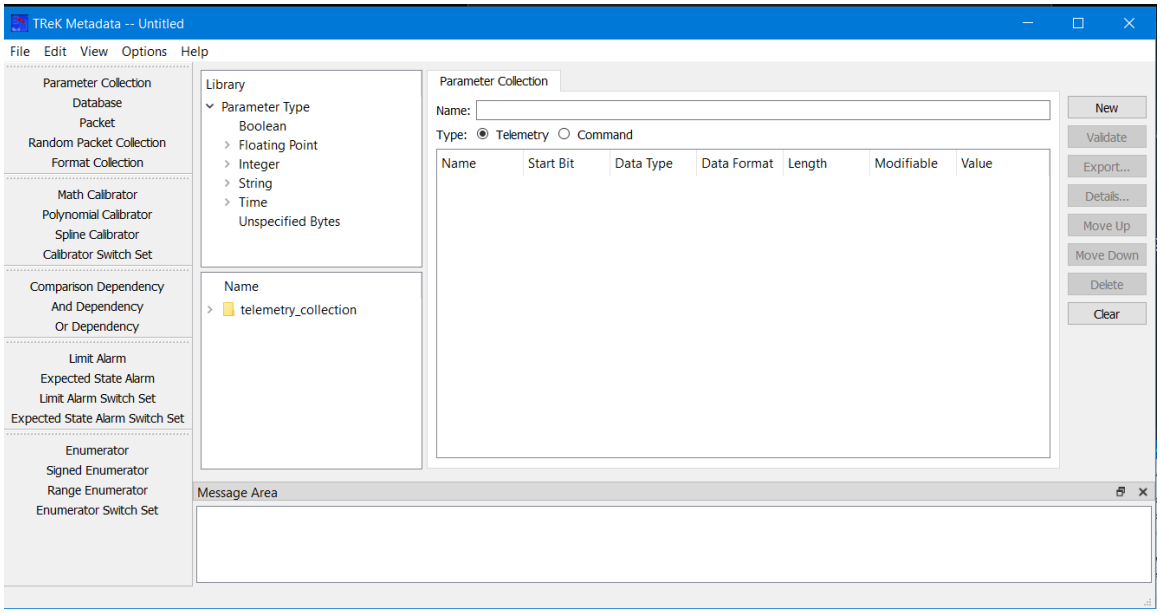


**Figure 4 Main Window in Parameter Collection Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to items you can use to create a Parameter Collection. The Parameter Type list provides a list of all the parameter types that can be added to a Parameter Collection. The lower half of the library area shows the Telemetry Collection or Command Collection list. It is configured to provide access to existing Parameter Collections stored in the trek_workspace. When you create a Parameter Collection and export it to the trek_workspace, it will automatically be added to the Parameter Collection list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Parameter Collection Tab. This is where you will define the details of your Parameter Collection. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Parameter Collection you are working on (e.g., Validate to validate the Parameter Collection or Export to export the Parameter Collection).

The Work Area provides the capability to define the following properties for the Parameter Collection:

Name
The name text field is used to enter the name of the Parameter Collection.

Type
The type selection is used to identify the type of Parameter Collection: Telemetry or Command.

Parameter List
The Parameter List is used to identify the parameters in the Parameter Collection.

The Metadata application uses Drag and Drop to move items from the Library Area into the Work Area.

To add a parameter, use your left mouse button to select a parameter from the Library Parameter Type list, and while holding the left mouse button down, drag the item to the Work Area and drop it as shown by the arrows in Figure 5.

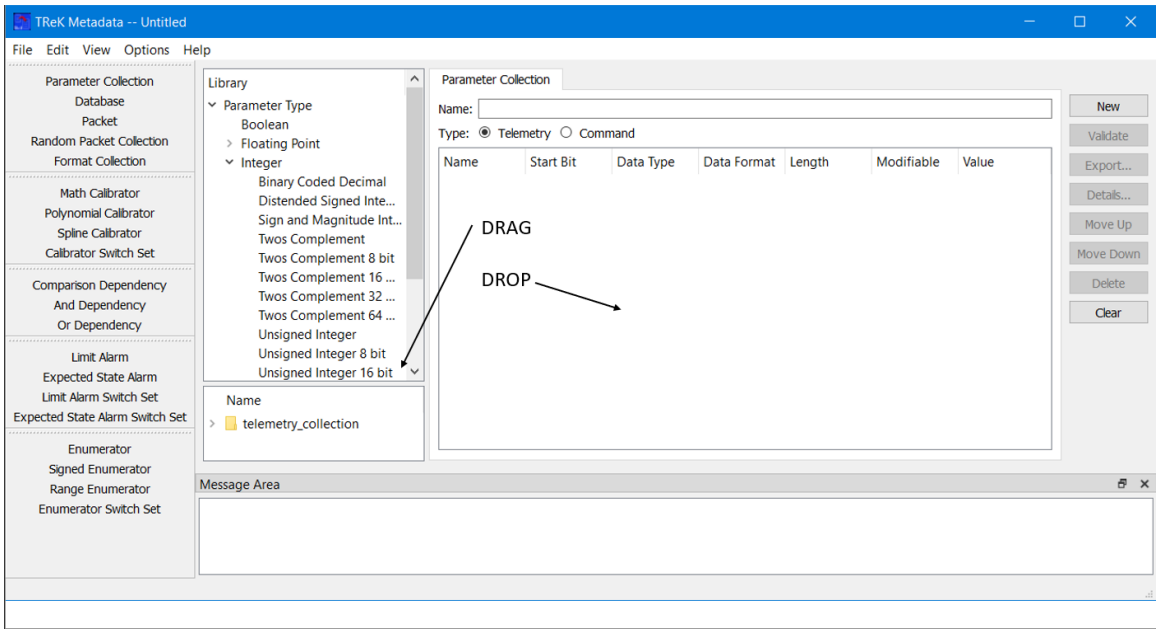


**Figure 5 Parameter Collection Drag and Drop**

After dropping the parameter, you should see the parameter in the list as shown in Figure 6.

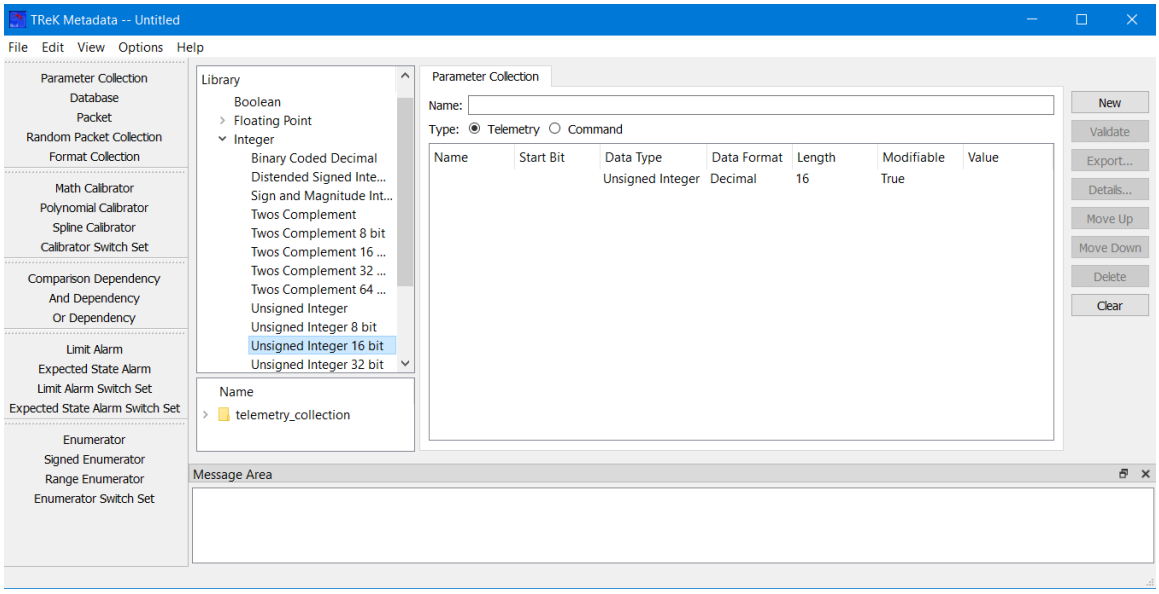


**Figure 6 Parameter Collection with Parameter Added**

To modify an existing Parameter Collection, you can drag the collection from the Telemetry Collection list or the Command Collection list and drop it in the Name field. This will load the Parameter Collection information into the Work Area.

Collection Drag and Drop Tips and Tricks

- If you drop an item from the Parameter Type list anywhere on the list it will be placed at the end of the list.
- If you drop an item from the Parameter Type list on top of another item, the item will be placed below that item in the list.
- If you drag an item from a Telemetry Collection list or Command Collection list and drop it in the Name field, the Parameter Collection information will be loaded into the Work Area.

The following functions are available when working with a Parameter Collection:

New
New provides the capability to clear the existing Parameter Collection information.

Validate
Validate provides the capability to validate the Parameter Collection information.

Export
Export provides the capability to save the Parameter Collection to a file.

Details
Details provides the capability to view and modify the properties associated with a selected parameter.

Move Up
Move Up provides the capability to move a selected parameter up in the list.

Move Down
Move Down provides the capability to move a selected parameter down in the list.

Delete
Delete provides the capability to delete a selected parameter.

Clear
Clear provides the capability to clear the Work Area.

## 6.2 Database

The Database configuration provides the capability to work with a TReK Database. Figure 7 shows the Main Window in the Database configuration. The main capabilities provided include creating a new database, viewing information about packets, collections, and types in the database, adding a packet to the database, deleting a packet from the database, and validating a database. To work with database tables directly, a

SQLite tool can be used to open and edit the database contents. There are many free and shareware tools available.

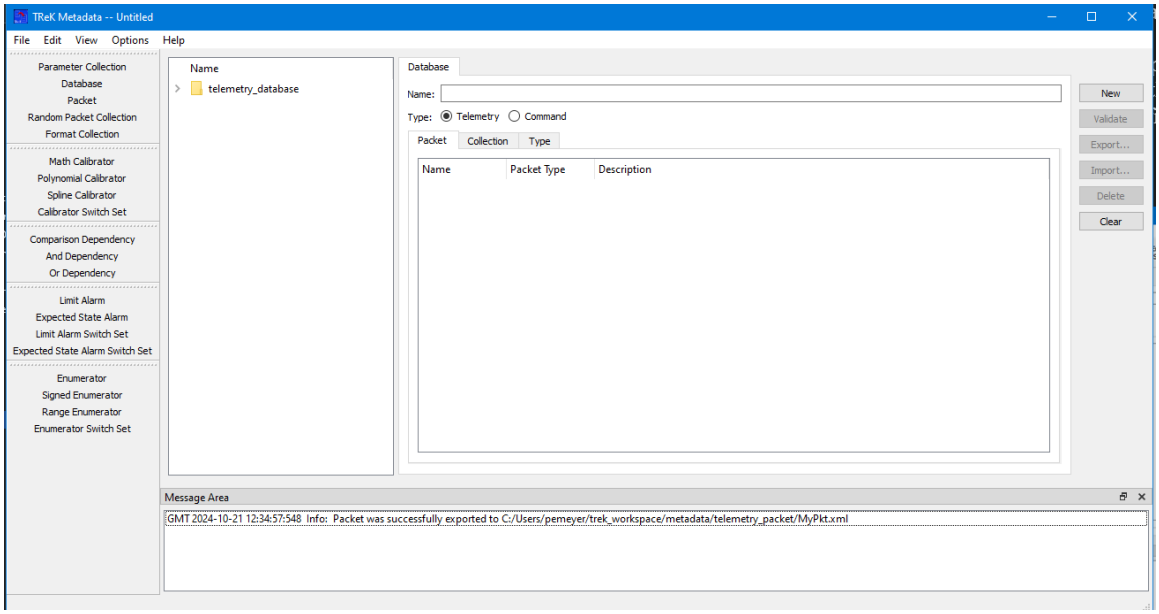


**Figure 7 Database Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to show a list of Telemetry Databases or Command Databases that reside in the trek_workspace. When you create a database and export it to the trek_workspace, it will automatically be added to the applicable database list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Database Tab. This is where you will see information stored in the Database. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Database you are working with (e.g., Validate to validate the Database or Export to export a packet from the database).

The Work Area provides the following information:

Name

The name field is used to display the name of the Database. The name field cannot be directly edited by typing in the field. The name field is populated when a new database is created using the New button or drag and drop is used to populate the field.

Type

The type selection is used to identify the type of Database: Telemetry or Command.

Packet Tab
The Packet Tab shows the packets in a Telemetry Database and the Commands in a Command Database.

Collection Tab
The Collection Tab shows the collections in a Telemetry Database and the collections in a Command Database.

Types Tab
The Types Tab shows the types in a Telemetry Database and the types in a Command Database.

The Metadata application uses Drag and Drop to move items from the Library Area into the Work Area.

To open a database, use your left mouse button to select a database from the Library, and while holding the left mouse button down, drag the item to the Name field and drop it. The database will be opened and the list of packets in the database will be displayed in the list as shown in Figure 8.

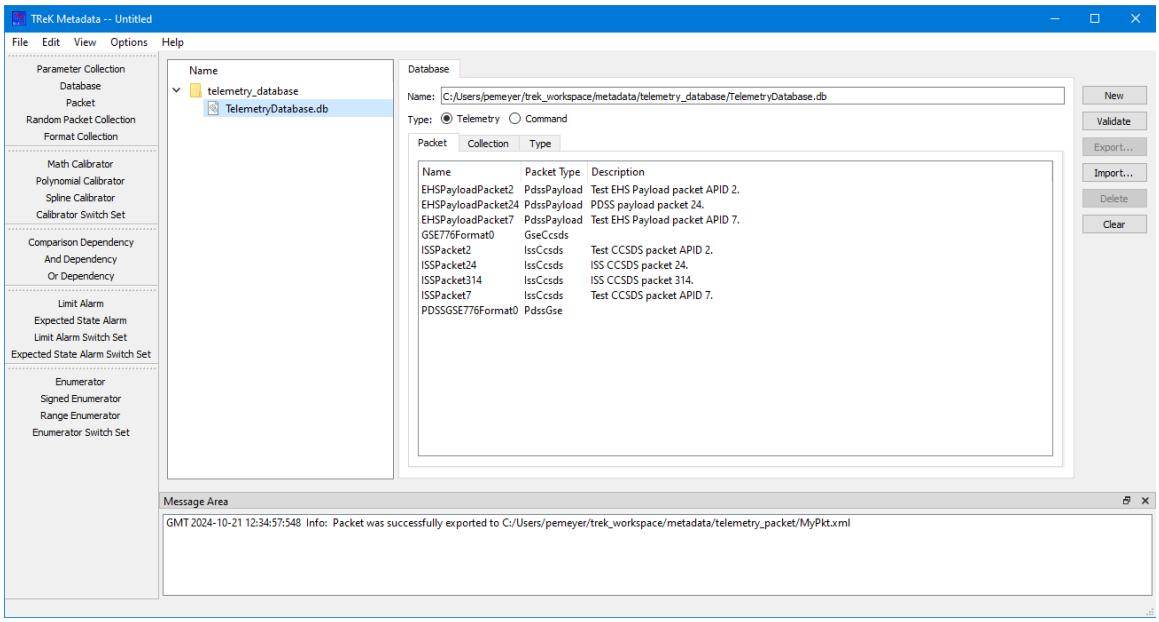


**Figure 8 Database Packets Shown In List**

The following functions are available when working with a Database:

New
New provides the capability to create a new database.

Validate
Validate provides the capability to validate a database.

Export
Export provides the capability to export a selected packet to a file.

Import
Import provides the capability to import information into a database. For a telemetry database, this includes the following types of imports: a packet definition defined in an ASCII file format as described in the TReK Telemetry Database Definition Document, a collection definition defined in an ASCII file format as described in the TReK Telemetry Database Definition Document, a type definition defined in an ASCII file format as described in the TReK Telemetry Database Definition Document, and a HOSC Ground Support Equipment (GSE) packet definition as described in the Payload to Generic User Interface Definition Document (PGUIDD).

Important Note about Parameter Names after Importing a GSE Packet Definition
When a GSE packet definition is imported the parameter names are changed. To keep parameter names unique, characters representing type of processing are added to the name. For example, suppose the name of the parameter is UGZG20RT2004J. If the parameter is unprocessed, the parameter will be named UGZG20RT2004J=UN where =UN was added to indicate that it is an unprocessed parameter. A converted value will have =CO added to the name, and a calibrated value will have =CA added to the name. Additional characters are also added to indicate Overall Status (=OStatus), Number of Samples (=Samples), and Sample Status Values (=Status). The following shows an example for a GSE parameter named UGZG20RT2004J:

| | |
|---|---|
| UGZG20RT2004J | Original Parameter name in GSE Packet Definition |
| UGZG20RT2004J=UN | Unprocessed Value Name |
| UGZG20RT2004J=UNOStatus | Unprocessed Overall Status Name |
| UGZG20RT2004J=UNSamples | Unprocessed Number of Samples Name |
| UGZG20RT2004J= UNStatus | Unprocessed Sample Status Values Name |

For a command database, this includes the following types of imports: a packet definition defined in an ASCII file format as described in the TReK Command Database Definition Document, a collection definition defined in an ASCII file format as described in the TReK Command Database Definition Document, and a type definition defined in an ASCII file format as described in the TReK Command Database Definition Document

Delete
Delete provides the capability to delete a selected packet from the database.

Clear
Clear provides the capability to clear the Work Area.

## 6.3 Packet

The Packet configuration provides the capability to work with a Packet. Figure 9 shows the Main Window in the Packet configuration.

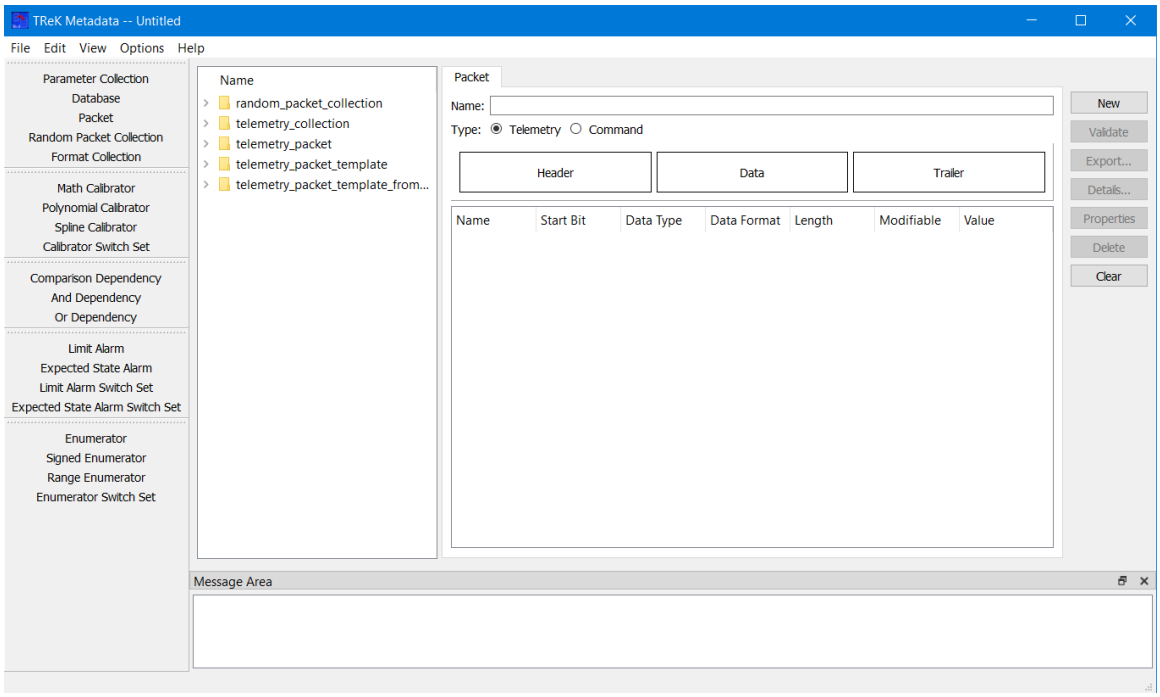


**Figure 9 Packet Configuration**

The Main Window Library Area is configured to provide access to items you can use to create or modify a packet. On the left-hand side of the Main Window you will see the Library Area. In the middle of the Main Window, the Work Area provides a place to define the details of your packet. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Packet you are working on. Each is described below.

Library Area

The Library Area is configured to provide access to items you can use to create or modify a Packet. The Telemetry Packet Template From Install list provides access to packet templates that are installed with the TReK software. You can create a packet from scratch. However, it is much easier to use a packet template and save yourself some work. There is also a Command Packet Template From Install. A packet template is a packet that includes pre-populated zones and packet properties corresponding to the specific type of packet. For example, a CCSDS packet template contains a CCSDS Header and the following packet properties: a Counter parameter, an APID Identifier, a Length parameter, and a Timestamp Parameter. By using a template, you can start your packet definition with all that information already defined. The Telemetry Collection or Command Collection list is configured to provide access to existing Parameter Collections stored in the trek_workspace. The Telemetry Packet or Command Packet list

is configured to provide access to existing Packets stored in the trek_workspace. When you create a Parameter Collection or Packet and export it to the trek_workspace, it will automatically be added to the Parameter Collection list or Packet list. The Random Packet Collection can be used to add a Random Packet Collection to a telemetry packet. The Telemetry Packet Template and Command Packet Template lists display user defined packet templates that are stored in the trek_workspace. The Library Area only displays items stored in the trek_workspace. If you need to work on an item outside of the trek_workspace, you will need to import the item using the appropriate configuration area.

Work Area

The Work Area provides a place to define the details of your packet. Notice the rectangle that contains three smaller rectangles with one for each zone of the Packet (Header, Data, and Trailer) as shown in Figure 10.

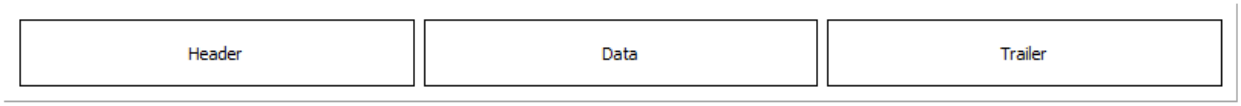


**Figure 10 Packet Drop Zones**

These rectangles are "Drop" zones. The large rectangle is a drop zone for the Entire Packet. If you drop a Packet Template or a Packet on the large rectangle, this will populate the entire Packet. If you drop a Packet or a Parameter Collection on one of the individual packet zones, that zone of the Packet will be updated. The drop zones will highlight so you can verify you are dropping the item in the correct location. Figure 11 shows what the "Entire Packet" drop zone looks like when it is highlighted. Figure 12 shows what the "Data Zone" drop zone looks like when it is highlighted.


**Figure 11 Entire Packet Drop Zone Highlighted**

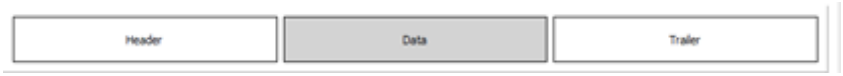


**Figure 12 Data Zone Drop Zone Highlighted**

Button Area

On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Packet you are working on (e.g., Validate to validate the Packet or Export to export the Packet).

The Work Area provides the capability to define the following properties for the Packet:

Name
The name field is used to enter the name of the Packet.

Type
The type selection is used to identify the type of Packet: Telemetry or Command.

Packet Contents List
The Packet Content list is used to show the contents of the Packet. Figure 13 shows an example of a Packet displayed in the Packet Contents List.

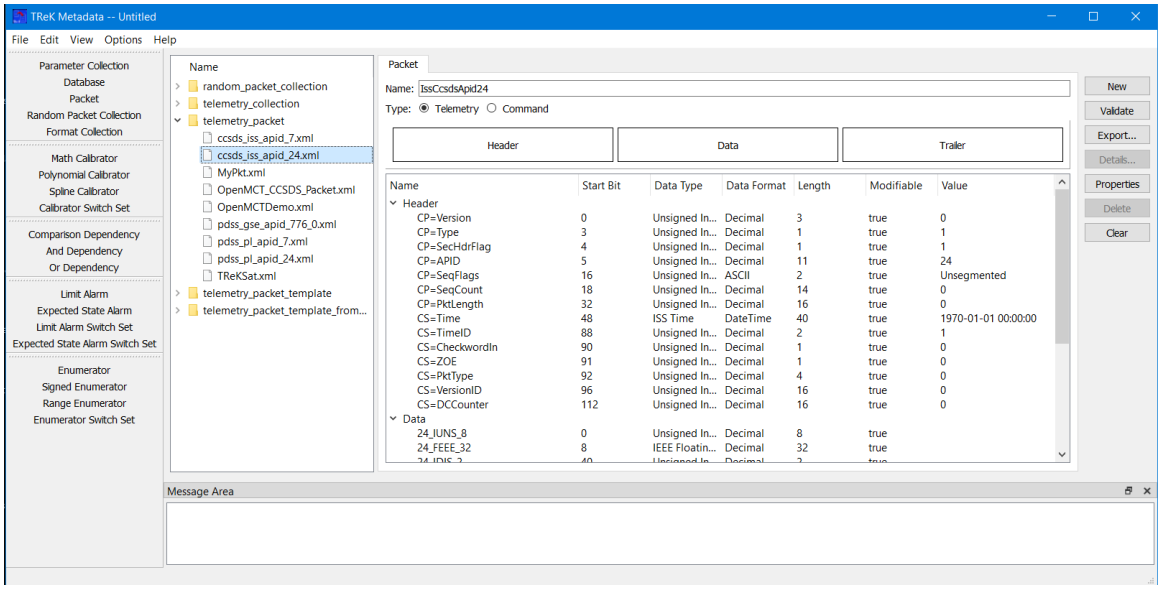


**Figure 13 Packet Contents**

The Metadata application uses Drag and Drop to move items from the Library Area into the Work Area.

To create or modify an existing Packet, you can drag items from the Library Area and drop them into one of the drop zones as described above. This will update the Packet information in the Work Area.

Packet Tips and Tricks

- If you drop a Packet Template or a Packet on the large rectangle, this will populate the entire Packet.
- If you drop a Packet or a Parameter Collection on one of the individual packet zones, that zone of the Packet will be updated.
- If you use your right mouse button you can access a context sensitive pop-up menu when selecting a Header Zone, Data Zone, or Trailer Zone in the list. This menu will provide capabilities such as Add Header, Add Data, Add Trailer, Replace, and Delete. This provides the capability to add, modify, or delete child zones.

The following functions are available when working with a Packet:

New
New provides the capability to clear the existing Packet information.

Validate
Validate provides the capability to validate the Packet information.

Export
Export provides the capability to save the Packet to a file.

Properties
Properties provides the capability to define Packet Properties. Please reference section 6.24 for details.

Details
Details provides more information about the selected item.

Delete
Delete provides the capability to delete a selected zone.

Clear
Clear provides the capability to clear the Work Area.

## 6.4 Random Packet Collection

The Random Packet Collection configuration provides the capability to work with a Random Packet Collection. Figure 14 shows the Main Window with the Random Packet Collection configuration. Random Packet Collections can be used to process data that is random such as ISS health and status.

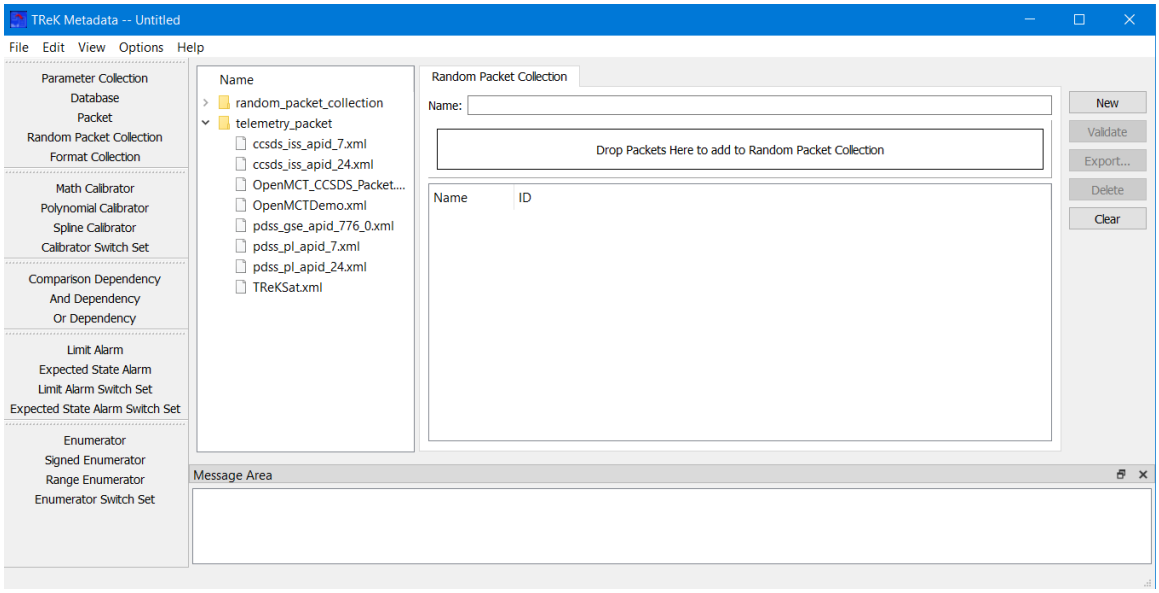


**Figure 14 Random Packet Collection**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to items you can use to create a Random Packet Collection. The Telemetry Packet list provides a list of telemetry packets that can be used to select a packet to add to a Random Packet Collection. The Random Packet Collection list is configured to provide access to existing Random Packet Collections stored in the trek_workspace. When you create a Random Packet Collection and export it to the trek_workspace, it will automatically be added to the Random Packet Collection list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Random Packet Collection Tab. This is where you will define the details of your Random Packet Collection. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Random Packet Collection you are working on (e.g., Validate to validate the Random Packet Collection, Export to export the Random Packet Collection, etc.).

The Work Area provides the capability to define the following properties for the Random Packet Collection:

Name
The name field is used to enter the name of the Random Packet Collection.

Packet Drop Zone
The Packet Drop Zone is used to add packets to the Random Packet Collection.

The Metadata application uses Drag and Drop to move items from the Library Area into the Work Area.

To add a packet to the Random Packet Collection, use your left mouse button to select a packet from the Library Telemetry Packet list, and while holding the left mouse button down, drag the item to the Packet Drop Zone and drop it. The packet will be added to the Random Packet Collection as shown in Figure 15.

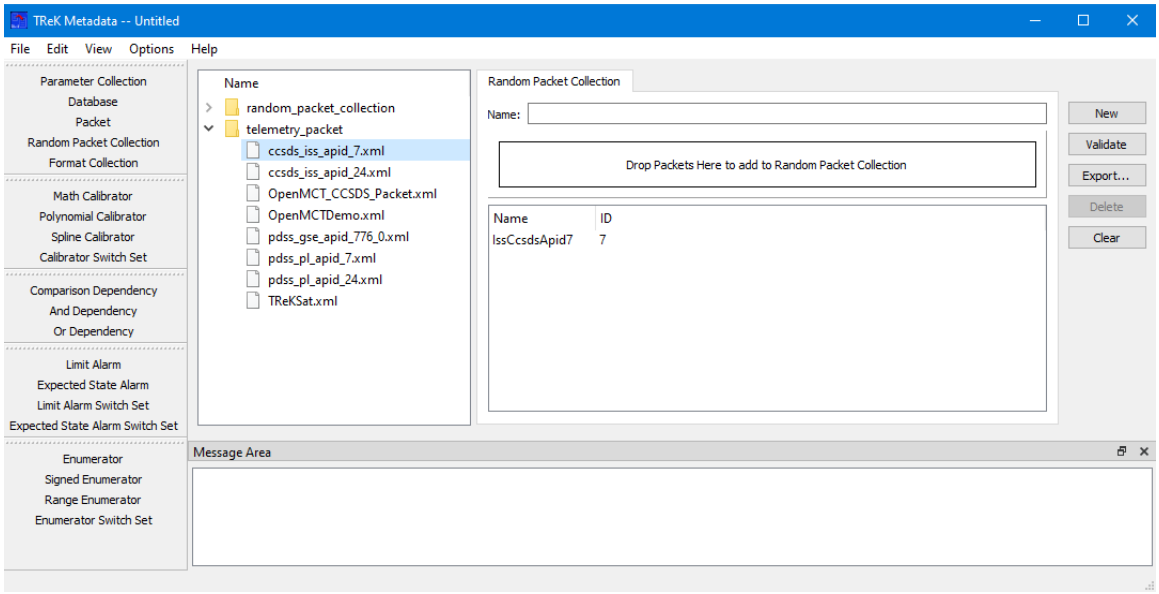


**Figure 15 Random Packet Collection with Packet**

Random Packet Collections require that all packets within it are of the same type (PDSS Payload, PDSS GSE, CCSDS, or blank, etc.). User created packets are blank type. The first packet placed in the Random Packet Collection sets the type for the collection. If you attempt to place a packet with a different type in the Random Packet Collection, it will be rejected.

The following functions are available when working with a Random Packet Collection:

New
New provides the capability to clear the existing Random Packet Collection information.

Validate
Validate provides the capability to validate the Random Packet Collection information.

Export
Export provides the capability to save the Random Packet Collection to a file.

Delete
Delete provides the capability to delete a selected packet in the Random Packet Collection.

Clear
Clear provides the capability to clear the Work Area.

## 6.5 Format Collection

The Format Collection configuration provides the capability to work with a Format Collection. Figure 16 shows the Main Window with the Format Collection configuration. Format Collections can be used to define a collection that contains multiple formats.

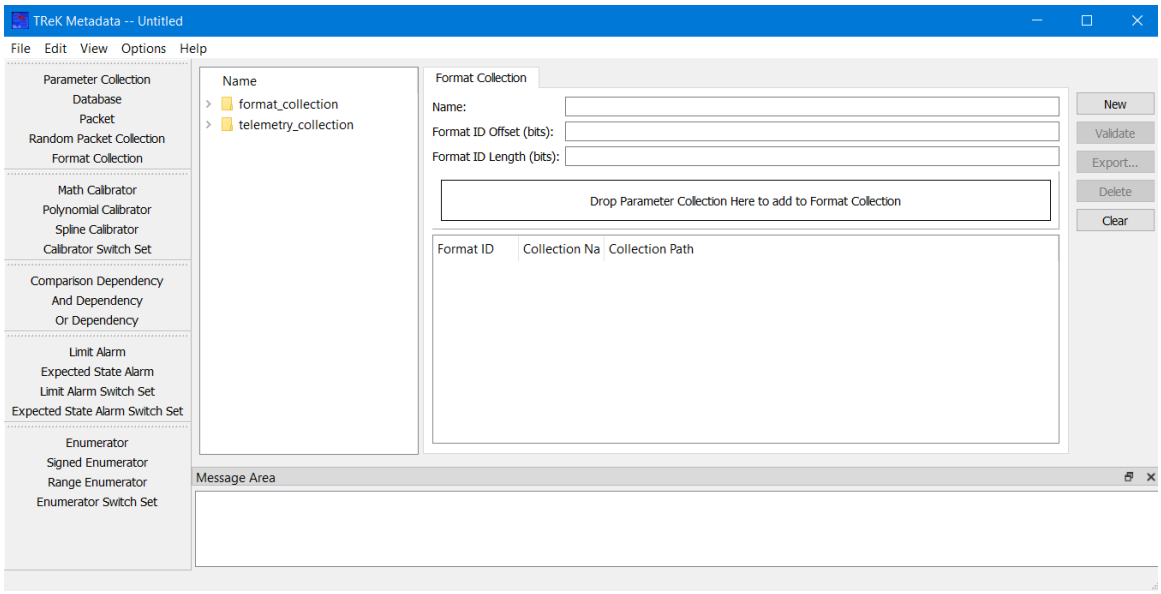


**Figure 16 Format Collection**

On the left hand side of the Main Window you will see the Library Area. It is configured to provide access to items you can use to create a Format Collection. The Telemetry Collection list provides a list of telemetry collections that can be used to select a collection to add to a Format Collection. The Format Collection list is configured to provide access to existing Format Collections stored in the trek_workspace. When you create a Format Collection and export it to the trek_workspace, it will automatically be added to the Format Collection list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Format Collection Tab. This is where you will define the details of your Format Collection. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Format Collection you are working on (e.g., Validate to validate the Format Collection, Export to export the Format Collection, etc.).

The Work Area provides the capability to define the following properties for the Format Collection:

Name
The name field is used to enter the name of the Format Collection.

Format ID Offset
The format ID offset field is used to enter the offset of the Format ID in the Format Collection.

Format ID Length
The format ID length field is used to enter the length of the Format ID in the Format Collection.

Collection Drop Zone
The Collection Drop Zone is used to add collections to the Format Collection.

The Metadata application uses Drag and Drop to move items from the Library Area into the Work Area.

To add a collection to the Format Collection, use your left mouse button to select a collection from the Telemetry Collection list. While holding the left mouse button down, drag the item to the Collection Drop Zone and drop it. The collection will be added to the Format Collection as shown in Figure 17. The Format ID Offset and Format ID Length fields must be populated with valid values before the collection can be added. Once the collection has been added to the list, the Format ID cell can be edited to enter the Format ID associated with the collection. The collection name will be the name that is defined in the collection and cannot be edited in the list. The collection path will be the absolute path to the collection file and cannot be edited in the list.

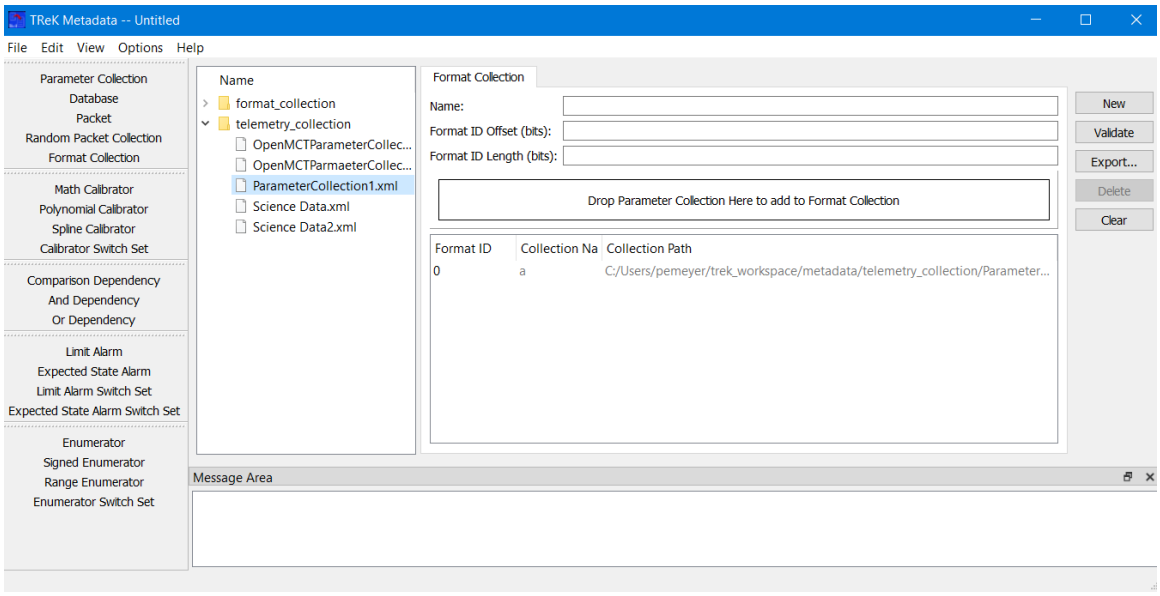


**Figure 17 Format Collection After Adding a Collection**

The following functions are available when working with a Format Collection:

New
New provides the capability to clear the existing Format Collection information.

Validate
Validate provides the capability to validate the Format Collection information.

Export
Export provides the capability to save the Format Collection to a file.

Delete
Delete provides the capability to delete a selected collection in the Format Collection.

Clear
Clear provides the capability to clear the Work Area.

## 6.6 Math Calibrator

The Math Calibrator configuration provides the capability to work with a Math Calibrator. Figure 18 shows the Main Window in the Math Calibrator configuration.

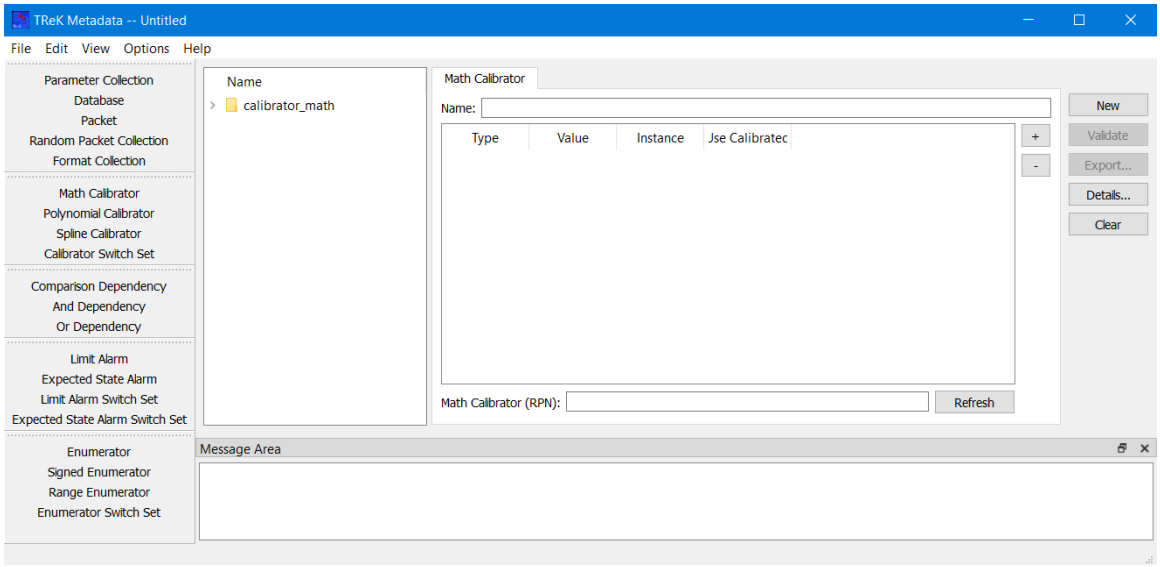


**Figure 18 Math Calibrator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Math Calibrators in the trek_workspace. When you create a Math Calibrator and export it to the trek_workspace, it will be added to the Math Calibrator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move

it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Math Calibrator Tab. This is where you will define the details of the Math Calibrator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Math Calibrator you are working on (e.g., Validate to validate the Math Calibrator or Export to export the Math Calibrator).

The Work Area provides the following information:

Name
The name field is used to display the name of the Math Calibrator.

Math Calibrator Components List
The Math Calibrator Components List is used to define the Math Calibrator. The Math Calibrator uses Reverse Polish Notation (RPN) and is limited to the following operators: +, -, *, /, ^, and ln. The Math Calibrator can use the value of this parameter, the converted or calibrated value of another parameter in the same packet, and constants. The + (plus) button is used to add a row to the list and the – (minus) button is used to delete a row from the list. Each row defines a component in the Math Calibrator equation. Figure 19 shows a populated Math Calibrator. The last row in the Math Calibrator should always be an Operand. The Math Calibrator (RPN) field is used to display the Math Calibrator equation in Reverse Polish Notation.

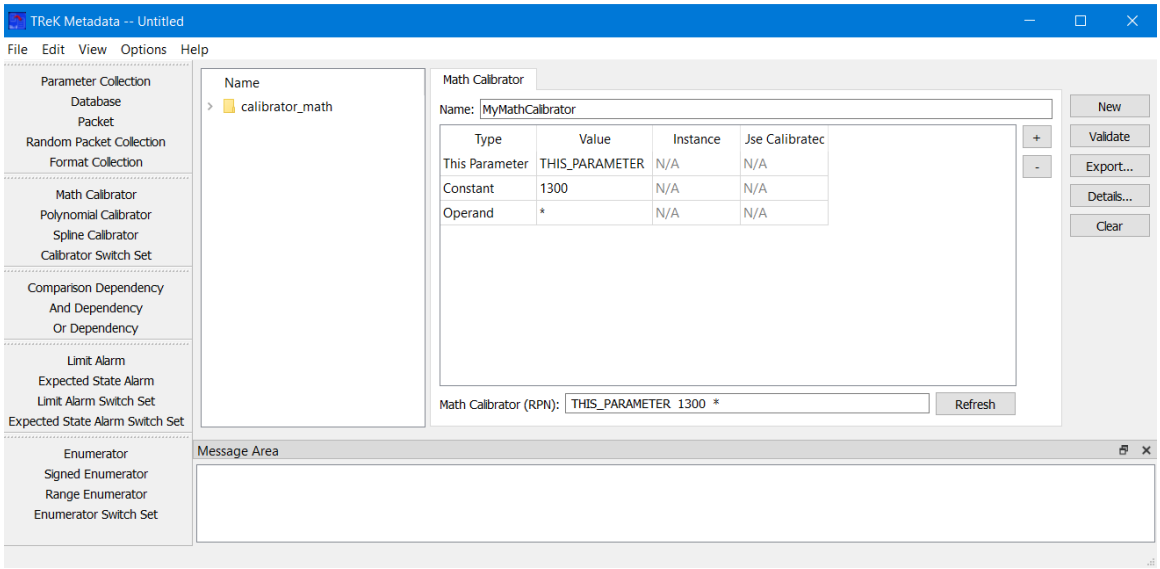


**Figure 19 Populated Math Calibrator**

After adding a blank row, the Type column provides a pulldown list of available types: The Parameter, Constant, Operand, or Other Parameter. The following table describes what should be entered in the Value, Instance, and Use Calibrated columns based on what Type is selected.

| Type | Value | Instance | Use Calibrated |
|---|---|---|---|
| This Parameter | Parameter Name | N/A | N/A |
| Constant | Numerical Value | N/A | N/A |
| Operand | Options: +, -, *, /, ^, ln | N/A | N/A |
| Other Parameter | Parameter Name | Instance of the Other Parameter | Yes or No to indicate if the Calibrated value of the other parameter should be used. |

Refresh
The Refresh button can be used to display the Math Calibrator in Reverse Polish Notation.

The following functions are available when working with a Math Calibrator:

New
New provides the capability to clear the existing Math Calibrator information.

Validate
Validate provides the capability to validate the Math Calibrator information.

Export
Export provides the capability to save the Math Calibrator to a file.

Details
Provides details about the Math Calibrator.

Clear
Clear provides the capability to clear the Work Area.

## 6.7 Polynomial Calibrator

The Polynomial Calibrator configuration provides the capability to work with a Polynomial Calibrator. Figure 20 shows the Main Window in the Polynomial Calibrator configuration.

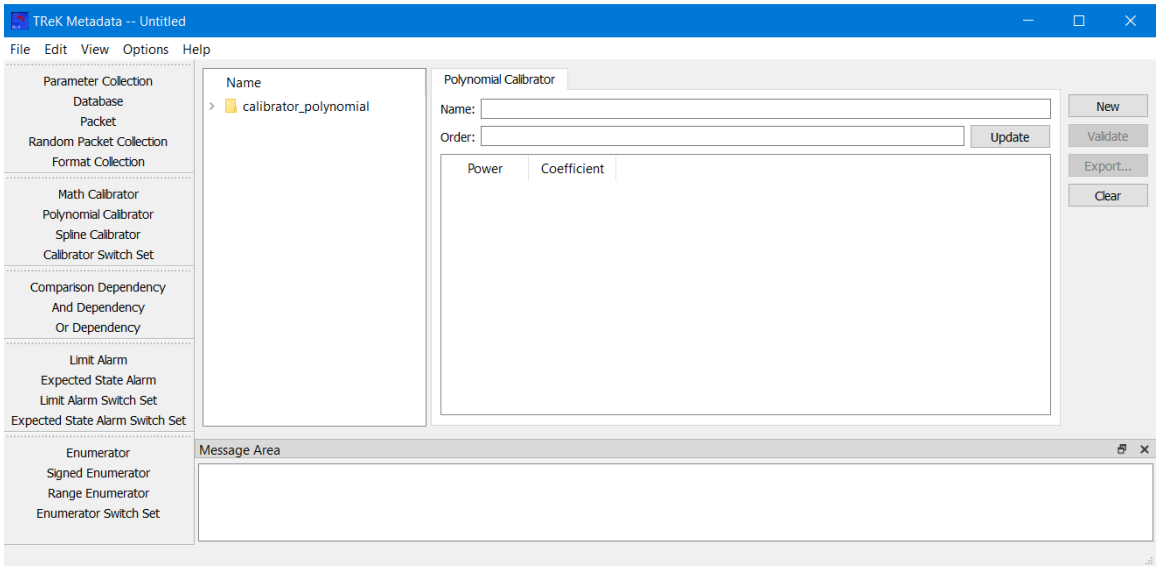


**Figure 20 Polynomial Calibrator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Polynomial Calibrators in the trek_workspace. When you create a Polynomial Calibrator and export it to the trek_workspace, it will be added to the Polynomial Calibrator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Polynomial Calibrator Tab. This is where you will define the details of the Polynomial Calibrator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Polynomial Calibrator you are working on (e.g., Validate to validate the Polynomial Calibrator or Export to export the Polynomial Calibrator).

The Work Area provides the following information:

Name
The name field is used to display the name of the Polynomial Calibrator.

Order
The order field is used to set the order of the Polynomial Calibrator. The correct number of rows is added to the Power/Coefficient List when the Update button is pushed.

Power Coefficient List
The Power Coefficient List is used to enter the Power and Coefficient information. The Power Coefficient List is configured when you enter a value for the Order and push the Update button. The results are shown in Figure 21.

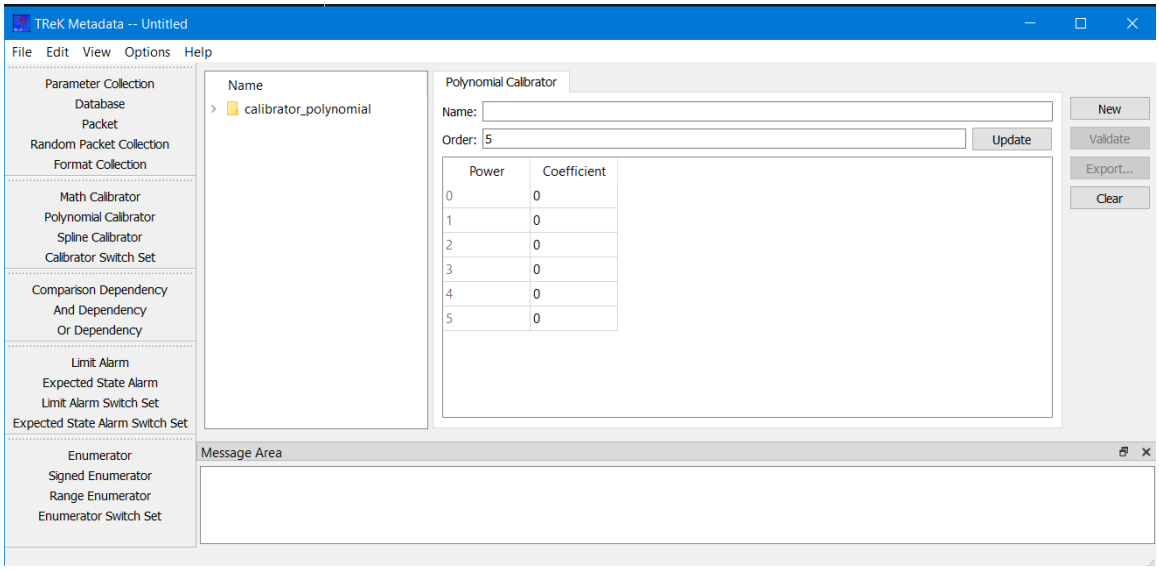


**Figure 21 Power Coefficient List Populated when Order is Set to 5**

The following functions are available when working with a Polynomial Calibrator:

New
New provides the capability to clear the existing Polynomial Calibrator information.

Validate
Validate provides the capability to validate the Polynomial Calibrator information.

Export
Export provides the capability to save the Polynomial Calibrator to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.8 Spline Calibrator

The Spline Calibrator configuration provides the capability to work with a Spline Calibrator. Figure 22 shows the Main Window in the Spline Calibrator configuration.

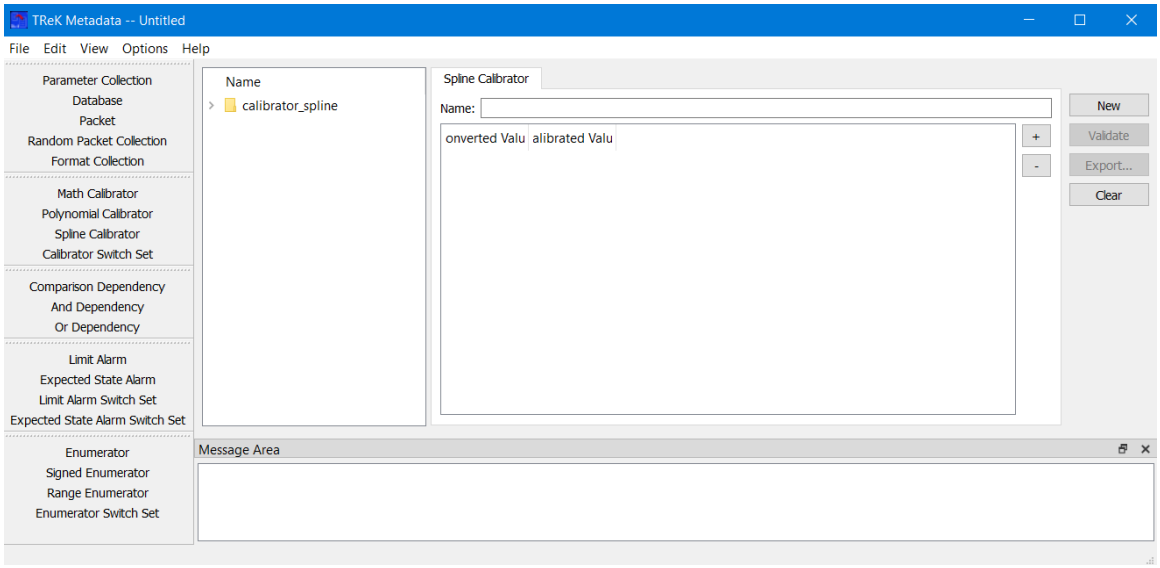


**Figure 22 Spline Calibrator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Spline Calibrators in the trek_workspace. When you create a Spline Calibrator and export it to the trek_workspace, it will be added to the Spline Calibrator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Spline Calibrator Tab. This is where you will define the details of the Spline Calibrator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Spline Calibrator you are working on (e.g., Validate to validate the Spline Calibrator or Export to export the Spline Calibrator).

The Work Area provides the following information:

Name
The name field is used to display the name of the Spline Calibrator.

Converted Value-Calibrated Value List
The Converted Value-Calibrated Value List is used to enter the Converted Value and Calibrated Value information. The + (plus) button is used to add a row to the list and the – (minus) button is used to delete a row from the list. Converted Value and Calibrated Value information can be entered directly into the list as shown in Figure 23.

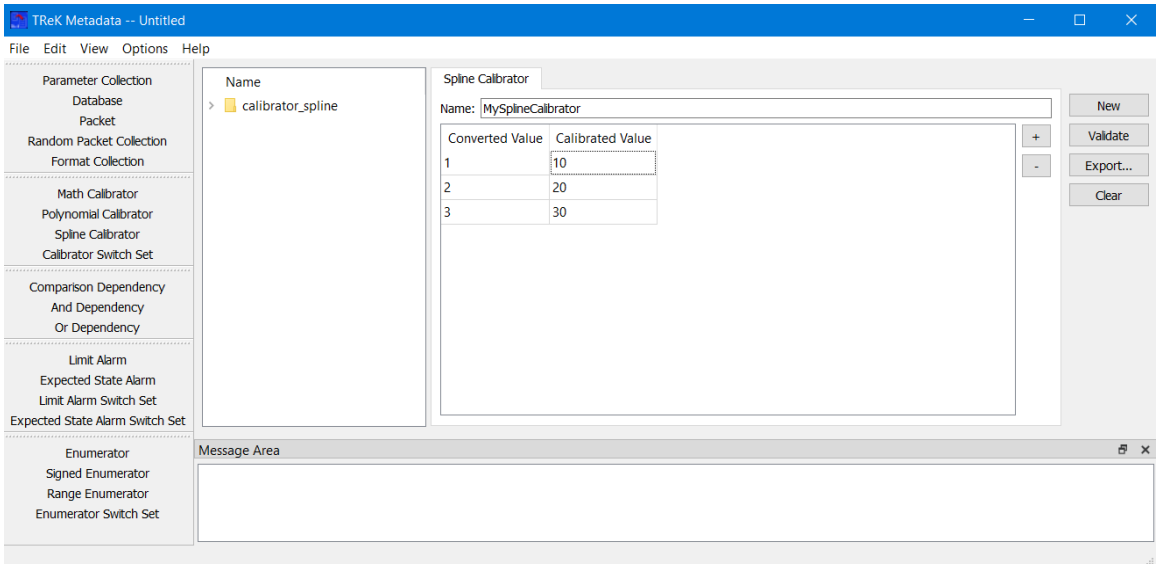


**Figure 23 Spline Calibrator Information**

The following functions are available when working with a Spline Calibrator:

New
New provides the capability to clear the existing Spline Calibrator information.

Validate
Validate provides the capability to validate the Spline Calibrator information.

Export
Export provides the capability to save the Spline Calibrator to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.9 Calibrator Switch Set

The Calibrator Switch Set configuration provides the capability to work with a Calibrator Switch Set. Figure 24 shows the Main Window in the Calibrator Switch Set configuration.

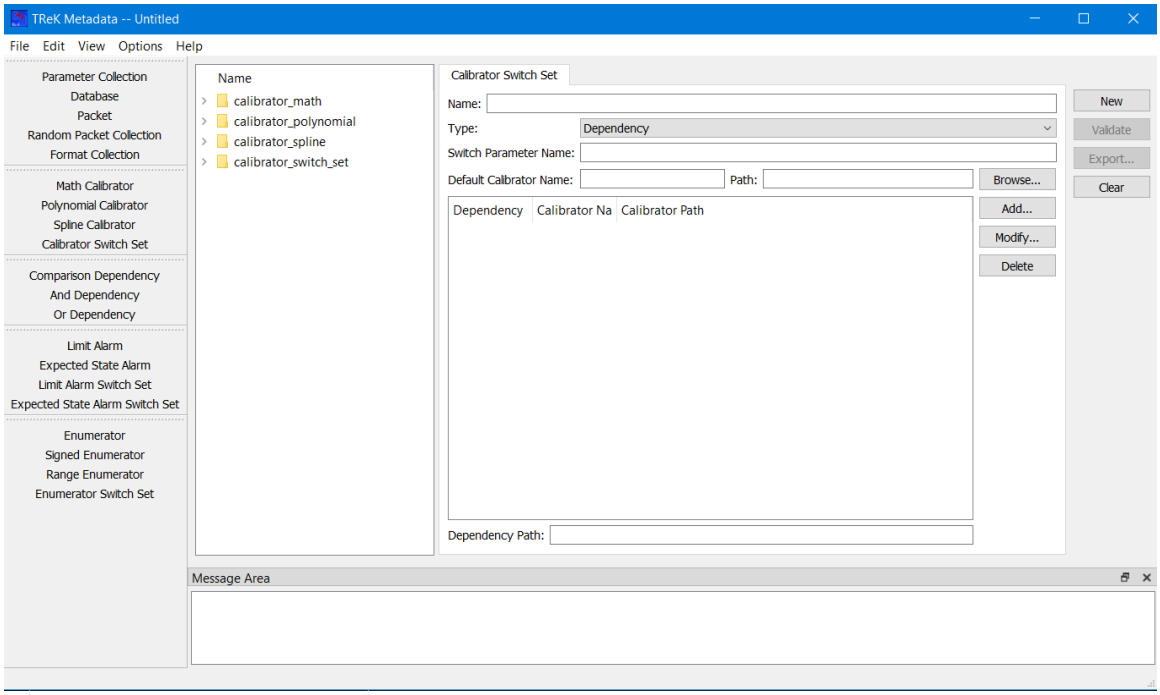


**Figure 24 Calibrator Switch Set Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Calibrators and Calibrator Switch Sets in the trek_workspace. When you create a Calibrator Switch Set and export it to the trek_workspace, it will be added to the Calibrator Switch Set list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Calibrator Switch Set Tab. This is where you will define the details of the Calibrator Switch Set. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Calibrator Switch Set you are working on (e.g., Validate to validate the Calibrator Switch Set or Export to export the Calibrator Switch Set).

The Work Area provides the following information:

Name
The name field is used to display the name of the Calibrator Switch Set.

Type
The type menu is used to specify the type of Calibrator Switch Set - Dependency, Enumerator or Range.

Switch Parameter Name
The parameter on which to switch.

Default Calibrator Name

The default calibrator name field is used to display the name of the default calibrator. This must be the name of a calibrator that exists in the trek_workspace. This field is not editable. It will be populated when you enter a path into the Default Calibrator Path field manually or by using the Browse button select a Calibrator.

Default Calibrator Path

The default calibrator path field is used to display the path of the default calibrator. The default calibrator must exist in the trek_workspace. You can enter the path manually or you can use the Browse button to browse for a Calibrator. If you delete the text in the Default Calibrator Path field, the Default Calibrator Name field will be cleared. If you enter information in the Default Calibrator Path field that is invalid, the Default Calibrator Name field will be cleared. If you enter valid information in the Default Calibrator Path field manually or by using the Browse button the Default Calibrator Name field will be automatically populated.

Set List

The Set List is used to enter set information for the Calibrator Switch Set. The Set List area will be configured based on switch type. The Add button is used to add a row to the list. The Modify button is used to modify an existing item in the list. Modify will be available when an item in the list is selected. Enumeration and Range (Low and High) information can be edited using the Modify dialog or directly in the list. Dependency information is derived from the Dependency identified in the Modify dialog and cannot be edited in the list. The Delete button is used to delete a row from the list. When a Calibrator is assigned to a set, the Calibrator name and Calibrator path will be displayed. When a Dependency row is selected, the Dependency path will be displayed in the Dependency Path field. Figure 25 shows a Dependency type switch. Figure 26 shows an Enumeration type switch. Figure 27 shows a Range type switch.

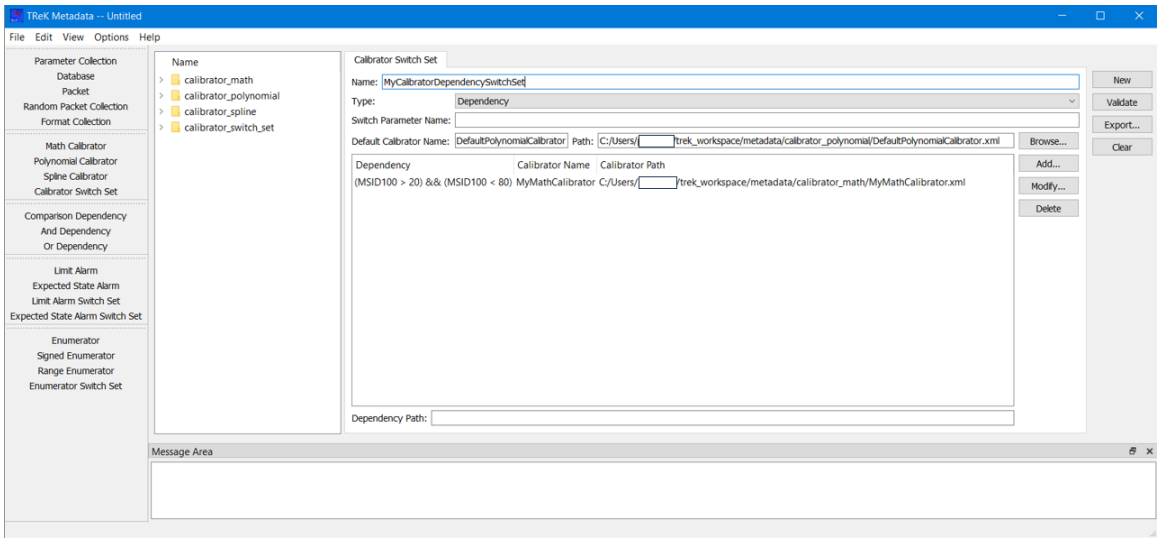


**Figure 25 Populated Calibrator Switch Set (Dependency)**

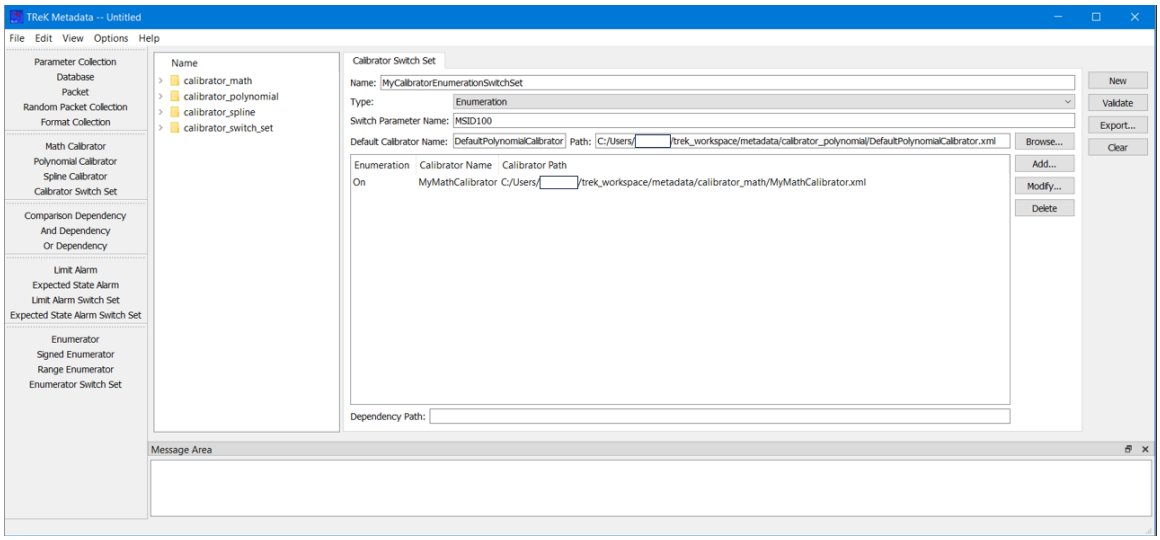


**Figure 26 Populated Calibrator Switch Set (Enumeration)**

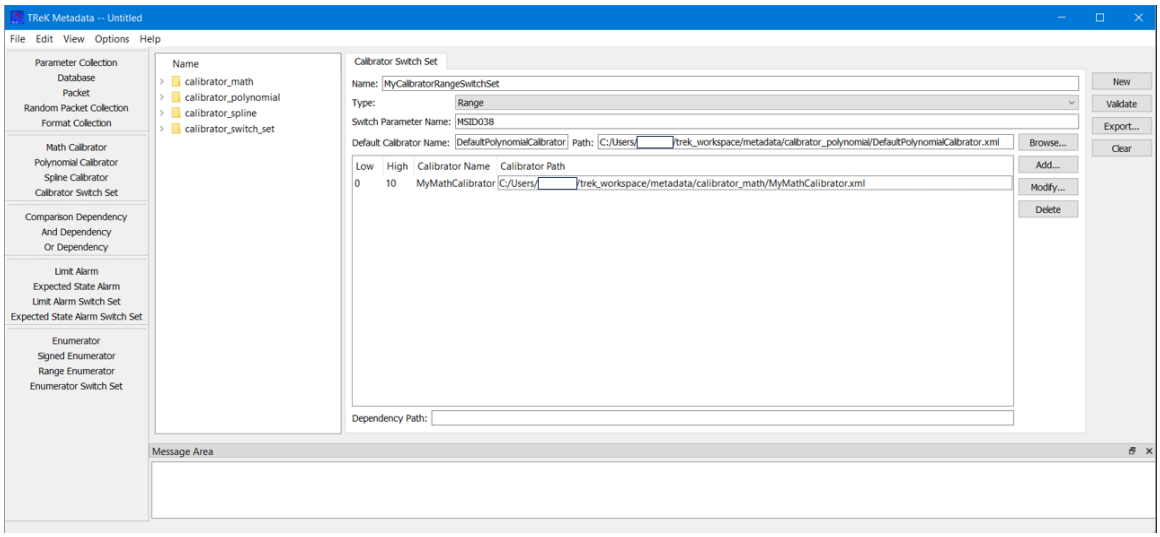


**Figure 27 Populated Calibrator Switch Set (Range)**

The following functions are available when working with a Calibrator Switch Set:

New
New provides the capability to clear the existing Calibrator Switch Set information.

Validate
Validate provides the capability to validate the Calibrator Switch Set information.

Export
Export provides the capability to save the Calibrator Switch Set to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.10 Comparison Dependency

The Comparison Dependency configuration provides the capability to work with a Comparison Dependency. Figure 28 shows the Main Window in the Comparison Dependency configuration.

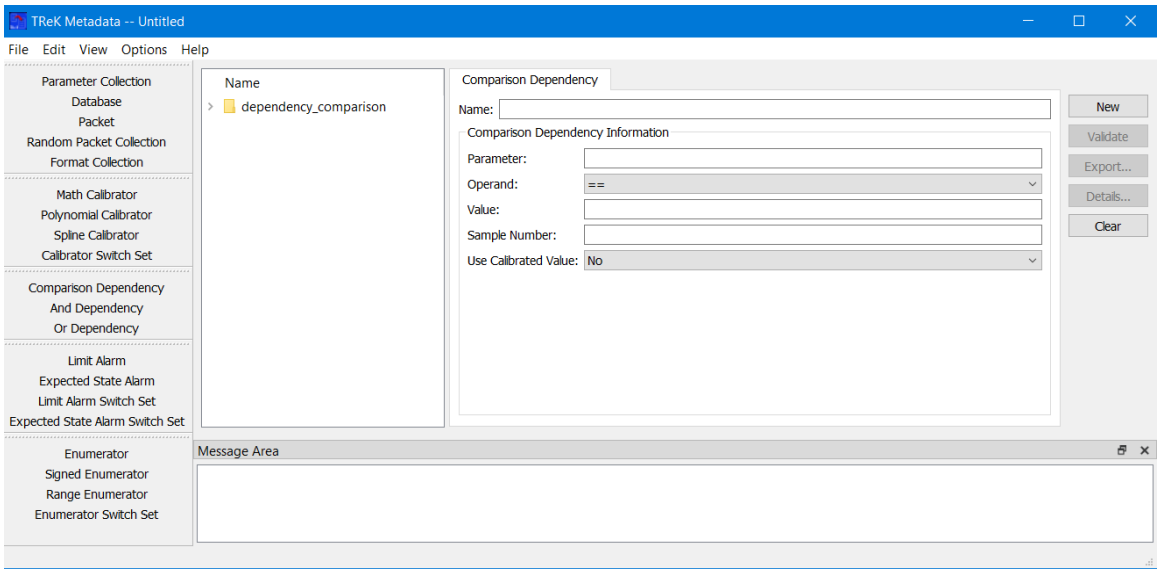


**Figure 28 Comparison Dependency Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Comparison Dependencies in the trek_workspace. When you create a Comparison Dependency and export it to the trek_workspace, it will be added to the Comparison Dependency list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Comparison Dependency Tab. This is where you will define the details of the Comparison Dependency. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Comparison Dependency you are working on (e.g., Validate to validate the Comparison Dependency or Export to export the Comparison Dependency).

The Work Area provides the following information:

Name
The name field is used to display the name of the Comparison Dependency.

Parameter
The name of the parameter whose value is checked.

Operand
The comparison type (==, !=, >, >=, <, <=).

Value
The value to compare against.

Sample Number
The sample number of the parameter to use. Sample numbers begin with 1.

Use Calibrated Value
Specifies whether the calibrated value should be used in the comparison.

Figure 29 Shows the Work Area after a Comparison Dependency was added to the list.

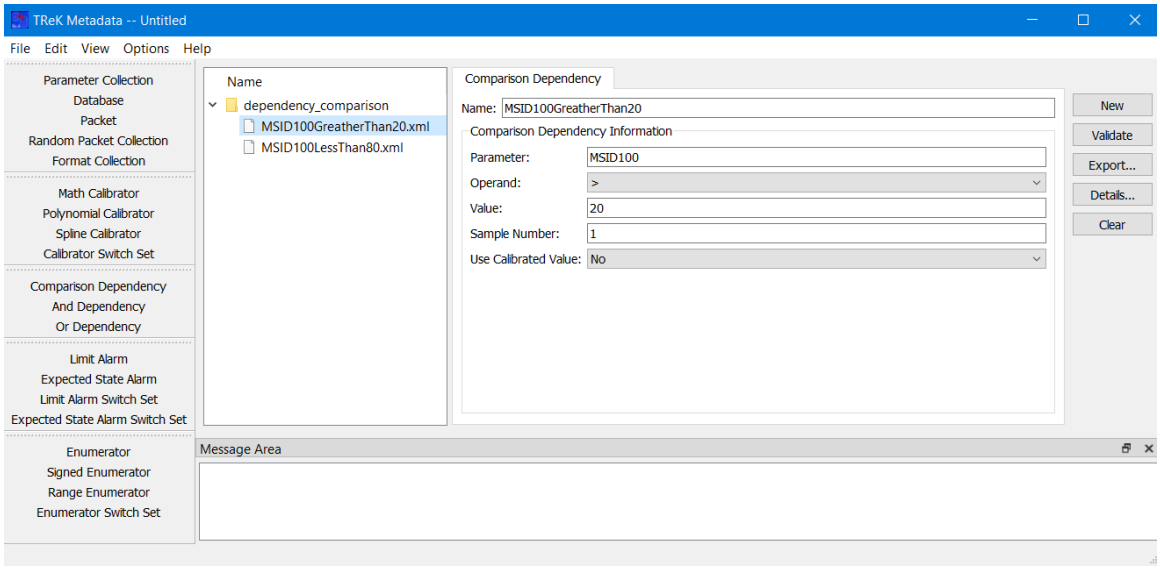


**Figure 29 Populated Comparison Dependency**

The following functions are available when working with a Comparison Dependency:

New
New provides the capability to clear the existing Comparison Dependency information.

Validate
Validate provides the capability to validate the Comparison Dependency information.

Export
Export provides the capability to save the Comparison Dependency to a file.

Details
Details provides more information about the Comparison Dependency.

Clear
Clear provides the capability to clear the Work Area.

## 6.11 And Dependency

The And Dependency configuration provides the capability to work with an And Dependency. Figure 30 shows the Main Window in the And Dependency configuration.

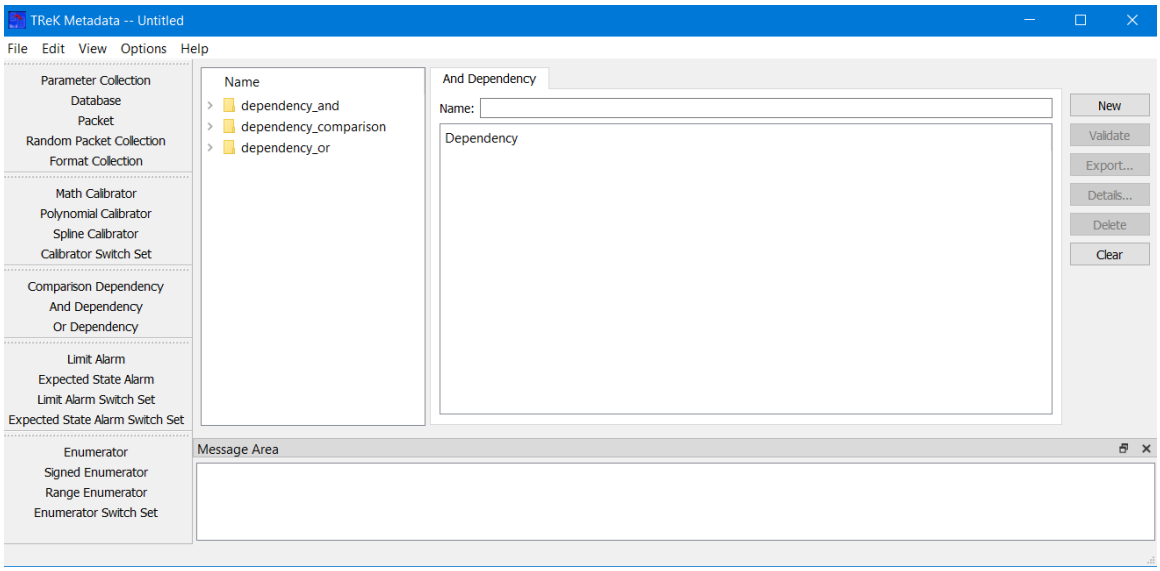


**Figure 30 And Dependency Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing And Dependencies, Comparison Dependencies, and Or Dependencies in the trek_workspace. When you create an And Dependency and export it to the trek_workspace, it will be added to the And Dependency list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the And Dependency Tab. This is where you will define the details of the And Dependency. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the And Dependency you are working on (e.g., Validate to validate the And Dependency or Export to export the And Dependency).

The Work Area provides the following information:

Name
The name field is used to display the name of the And Dependency.

Dependency
The Dependency List is used to define the components of the And Dependency. To define an And Dependency, drag and drop items from the Library into the Dependency list in the work area. Figure 31 shows the Dependency List after two Comparison Dependencies were added to the list.

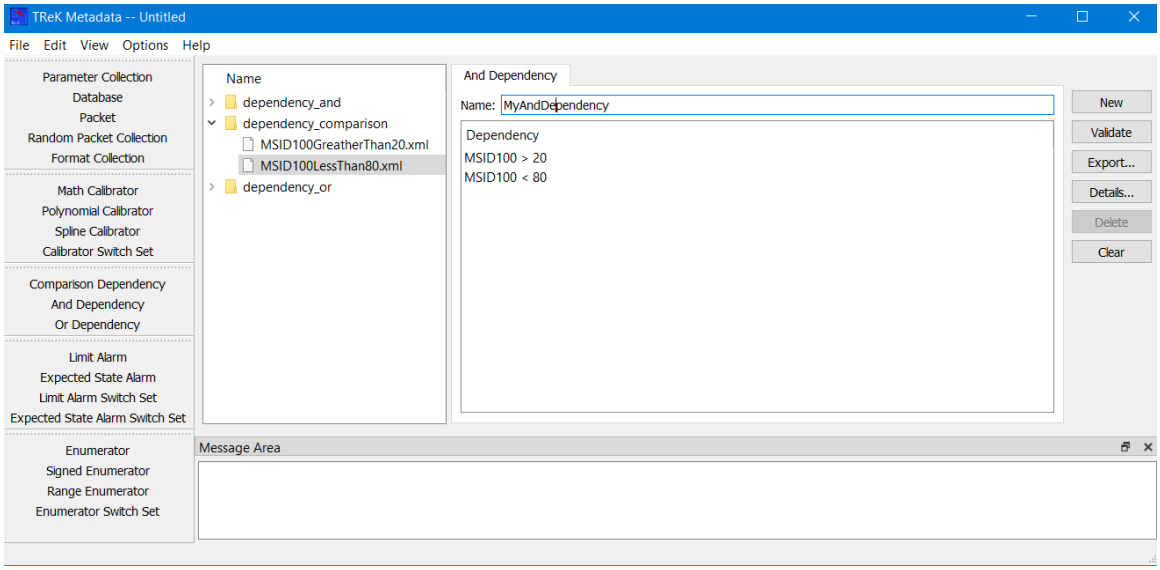


**Figure 31 Populated And Dependency**

The following functions are available when working with an And Dependency:

New
New provides the capability to clear the existing And Dependency information.

Validate
Validate provides the capability to validate the And Dependency information.

Export
Export provides the capability to save the And Dependency to a file.

Details
Details provides more information about the And Dependency.

Delete
Delete provides the capability to delete a selected dependency from the list.

Clear
Clear provides the capability to clear the Work Area.

## 6.12 Or Dependency

The Or Dependency configuration provides the capability to work with an Or Dependency. Figure 32 shows the Main Window in the Or Dependency configuration.

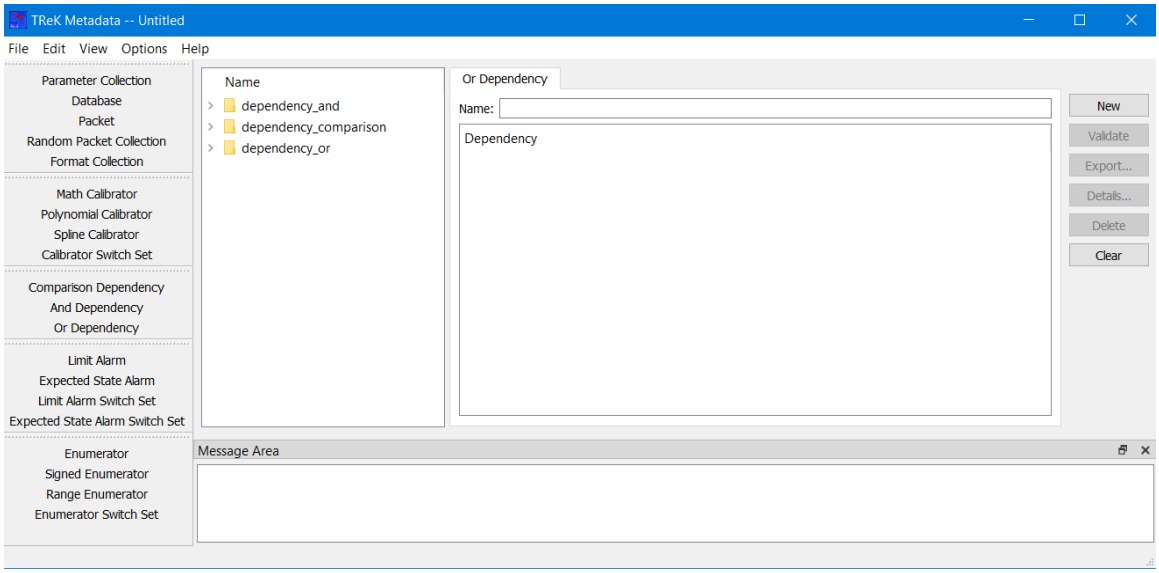


**Figure 32 Or Dependency Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing And Dependencies, Comparison Dependencies, and Or Dependencies in the trek_workspace. When you create an Or Dependency and export it to the trek_workspace, it will be added to the Or Dependency list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Or Dependency Tab. This is where you will define the details of the Or Dependency. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Or Dependency you are working on (e.g., Validate to validate the Or Dependency or Export to export the Or Dependency).

The Work Area provides the following information:

Name

The name field is used to display the name of the Or Dependency.

Dependency

The Dependency List is used to define the components of the Or Dependency. To define an Or Dependency, drag and drop items from the Library into the Dependency list in the

work area. Figure 33 shows the Dependency List after two Comparison Dependencies were added to the list.

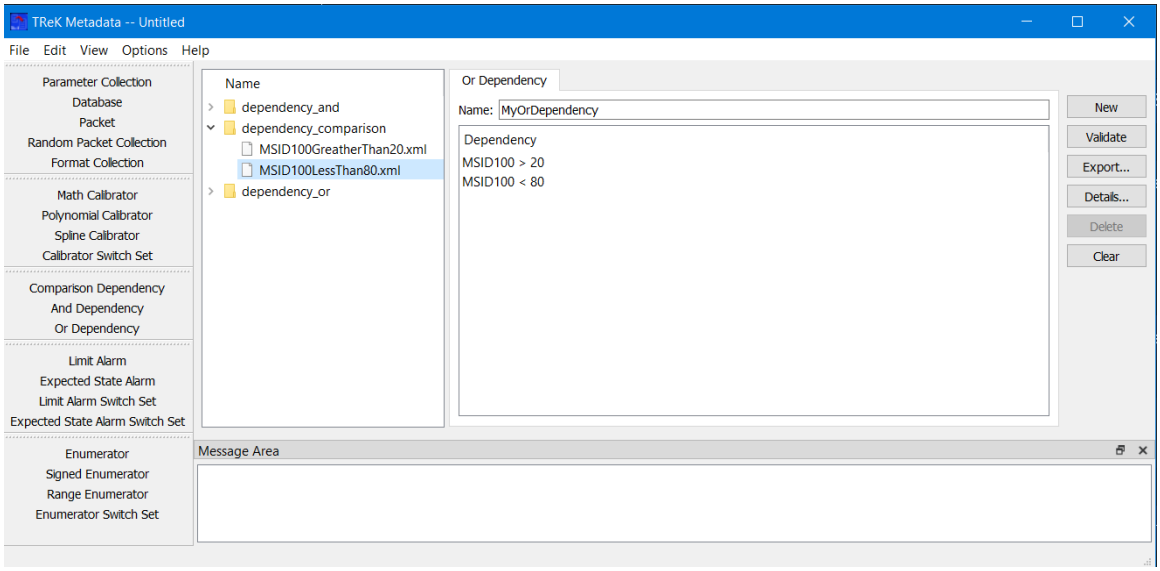


**Figure 33 Populated Or Dependency**

The following functions are available when working with an Or Dependency:

New
New provides the capability to clear the existing Or Dependency information.

Validate
Validate provides the capability to validate the Or Dependency information.

Export
Export provides the capability to save the Or Dependency to a file.

Details
Details provides more information about the Or Dependency.

Delete
Delete provides the capability to delete a selected Or Dependency from the list.

Clear
Clear provides the capability to clear the Work Area.

## 6.13 Limit Alarm

The Limit Alarm configuration provides the capability to work with a Limit Alarm. Figure 34 shows the Main Window in the Limit Alarm configuration.

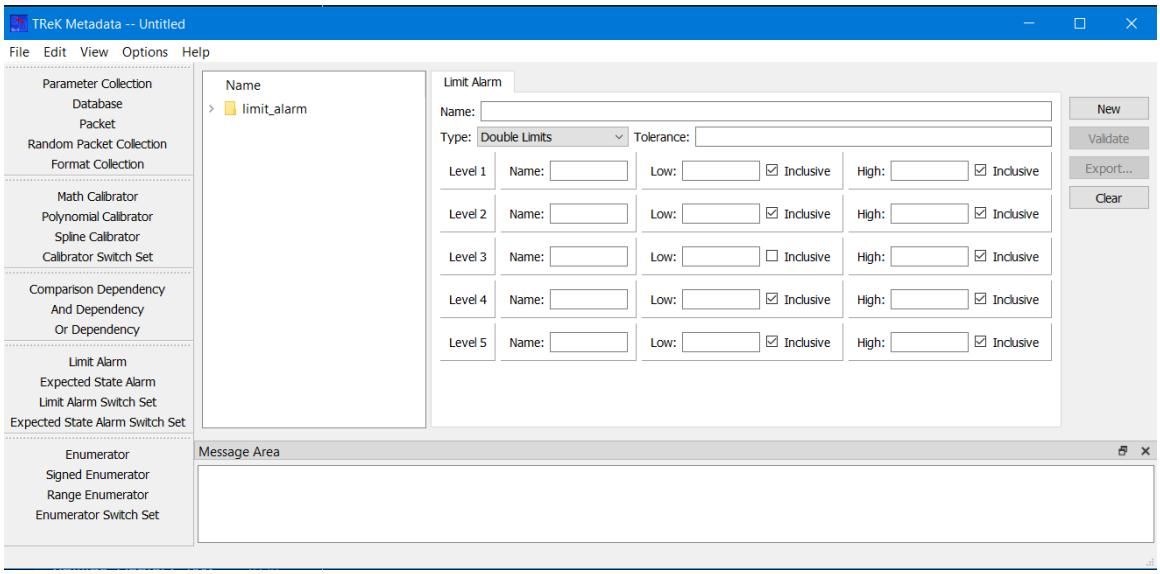


**Figure 34 Limit Alarm Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Limit Alarms in the trek_workspace. When you create a Limit Alarm and export it to the trek_workspace, it will be added to the Limit Alarm list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Limit Alarm Tab. This is where you will define the details of the Limit Alarm. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Limit Alarm you are working on (e.g., Validate to validate the Limit Alarm or Export to export the Limit Alarm).

The Work Area provides the following information:

Name
The name field is used to display the name of the Limit Alarm.

Type
The type menu is used to select the type of Limit Alarm. You can choose Double Limits, Integer Limits or Unsigned Integer Limits.

Tolerance
The number of times a value must be outside the current state before transitioning to the new state.

Limit Alarm Levels

Fields are provided to set details for each Limit Alarm Level as shown in Figure 35. If a high or low value is provided the Level is considered in use. The Low Inclusive checkbox is used to indicate if the Low value is included in the check. The High Inclusive checkbox is used to indicate if the High value is included in the check.


**Figure 35 Limit Alarm Level**

The following functions are available when working with a Limit Alarm:

New

New provides the capability to clear the existing Limit Alarm information.

Validate

Validate provides the capability to validate the Limit Alarm information.

Export

Export provides the capability to save the Limit Alarm to a file.

Clear

Clear provides the capability to clear the Work Area.

## 6.14 Expected State Alarm

The Expected State Alarm configuration provides the capability to work with an Expected State Alarm. Figure 36 shows the Main Window in the Expected State Alarm configuration.

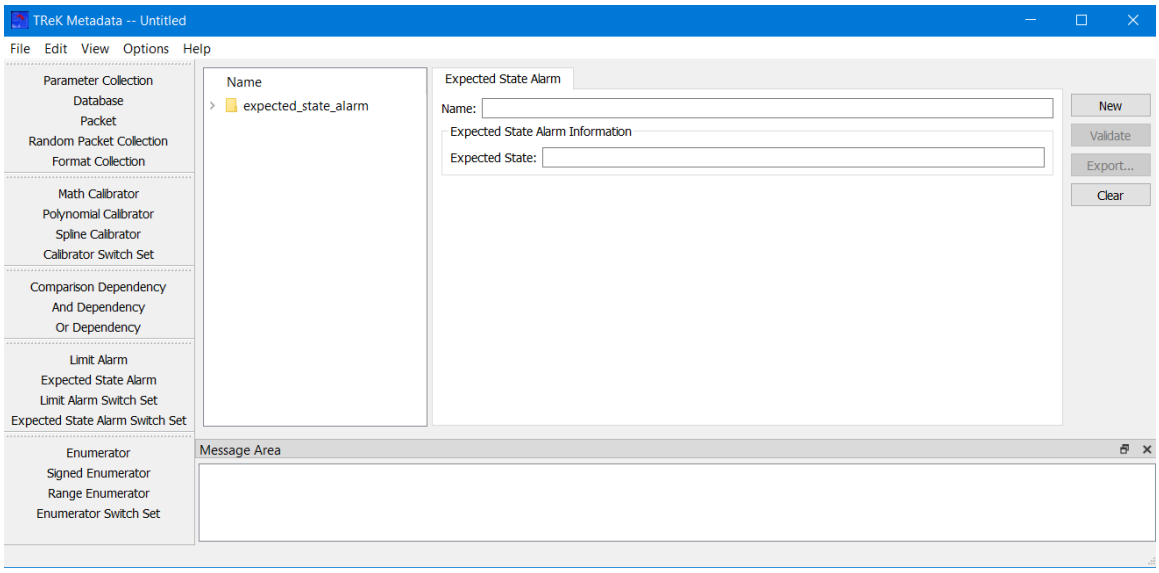


**Figure 36 Expected State Alarm Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Expected State Alarms in the trek_workspace. When you create an Expected State Alarm and export it to the trek_workspace, it will be added to the Expected State Alarm list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Expected State Alarm Tab. This is where you will define the details of the Expected State Alarm. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Expected State Alarm you are working on (e.g., Validate to validate the Expected State Alarm or Export to export the Expected State Alarm).

The Work Area provides the following information:

Name
The name field is used to display the name of the Expected State Alarm.

Expected State
The expected state field is used to enter the expected state for the Expected State Alarm.

The following functions are available when working with an Expected State Alarm:

New
New provides the capability to clear the existing Expected State Alarm information.

Validate
Validate provides the capability to validate the Expected State Alarm information.

Export
Export provides the capability to save the Expected State Alarm to a file.

Clear
Clear provides the capability to clear the Work Area.

### 6.15 Limit Alarm Switch Set

The Limit Alarm Switch Set configuration provides the capability to work with a Limit Alarm Switch Set. Figure 37 shows the Main Window in the Limit Alarm Switch Set configuration.

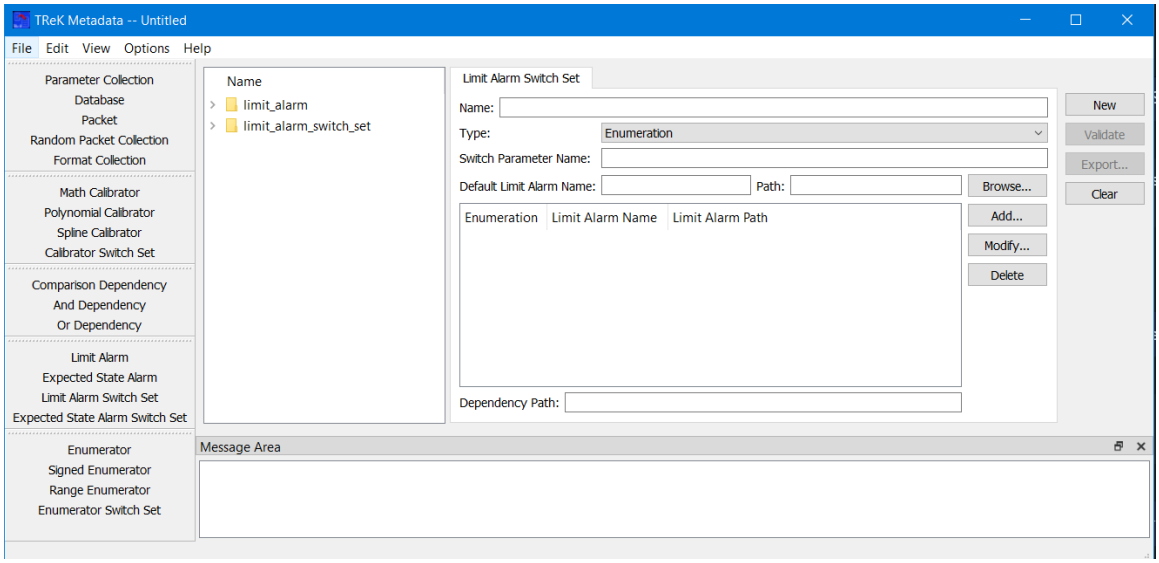


**Figure 37 Limit Alarm Switch Set Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Limit Alarms and Limit Alarm Switch Sets in the trek_workspace. When you create a Limit Alarm Switch Set and export it to the trek_workspace, it will be added to the Limit Alarm Switch Set list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Limit Alarm Switch Set Tab. This is where you will define the details of the Limit Alarm Switch Set. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Limit Alarm Switch Set you are working on (e.g., Validate to validate the Limit Alarm Switch Set or Export to export the Limit Alarm Switch Set).

The Work Area provides the following information:

Name
The name field is used to display the name of the Limit Alarm Switch Set.

Type
The type pull-down menu is used to specify the type of Limit Alarm Switch Set – Dependency, Enumeration or Range.

Switch Parameter Name
The name of the parameter on which to switch.

Default Limit Alarm Name
The default limit alarm name field is used to display the name of the default limit alarm. This must be the name of a limit alarm that exists in the trek_workspace. This field is not

editable. It will be populated when you enter a path into the Default Limit Alarm Path field manually or by using the Browse button to select a Limit Alarm.

Default Limit Alarm Path

The default limit alarm path field is used to display the path of the default limit alarm. The default limit alarm must exist in the trek_workspace. You can enter the path manually or you can use the Browse button to browse for a Limit Alarm. If you delete the text in the Default Limit Alarm Path field, the Default Limit Alarm Name field will be cleared. If you enter information in the Default Limit Alarm Path field that is invalid, the Default Limit Alarm Name field will be cleared. If you enter valid information in the Default Limit Alarm Path field manually or by using the Browse button, the Default Limit Alarm Name field will be automatically populated.

Set List

The Set List area is used to enter set information for the Limit Alarm Switch Set. The Set List area will be configured based on switch type. The Add button is used to add a row to the list. The Modify button is used to modify an existing item in the list. Modify will be available when an item in the list is selected. Enumeration and Range (Low and High) information can be edited using the Modify dialog or directly in the list. Dependency information is derived from the Dependency identified in the Modify dialog and cannot be edited in the list. The Delete button is used to delete a row from the list. When a Limit Alarm is assigned to a set, the Limit Alarm name and Limit Alarm path will be displayed. When a Dependency row is selected, the Dependency path will be displayed in the Dependency Path field. Figure 38 shows a Dependency type switch. Figure 39 shows an Enumeration type switch. Figure 40 shows a Range type switch.

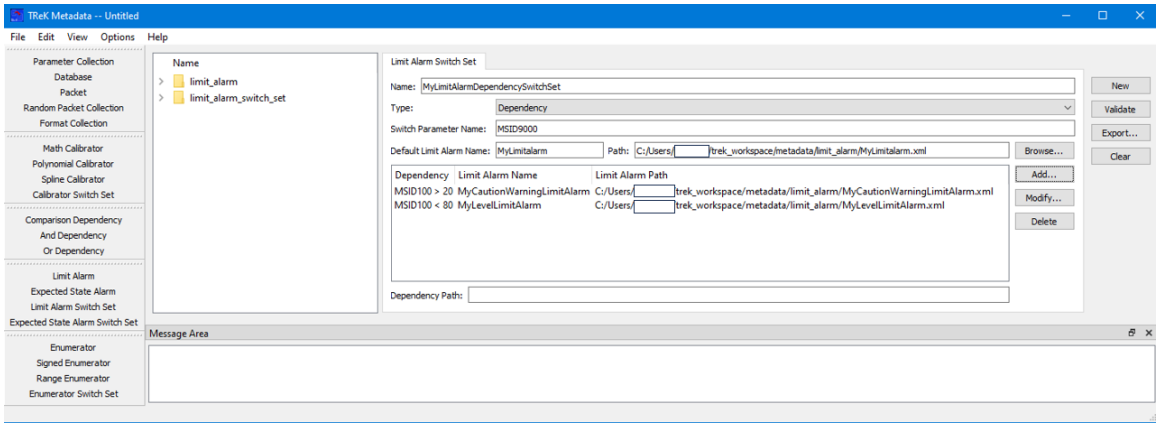


**Figure 38 Populated Limit Alarm Switch Set (Dependency)**

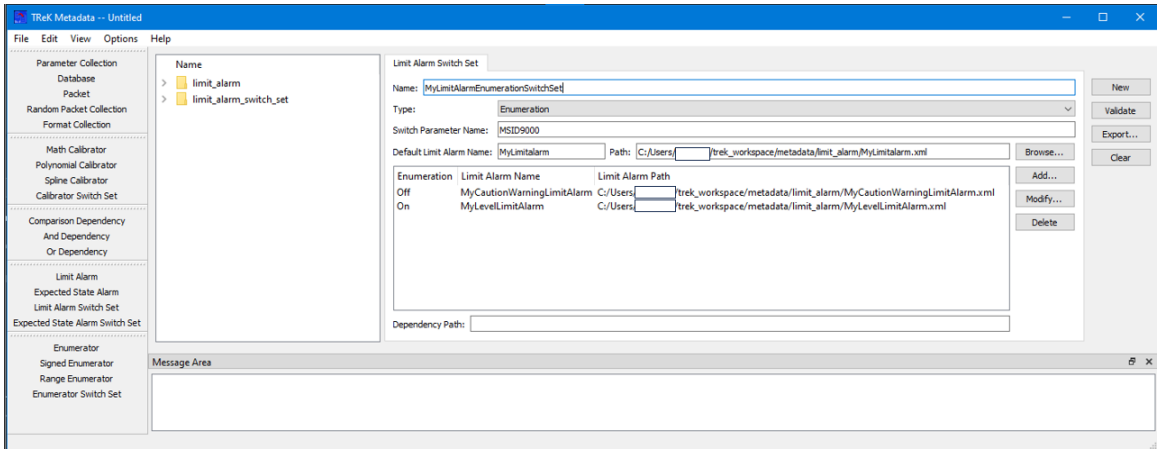


**Figure 39 Populated Limit Alarm Switch Set (Enumeration)**

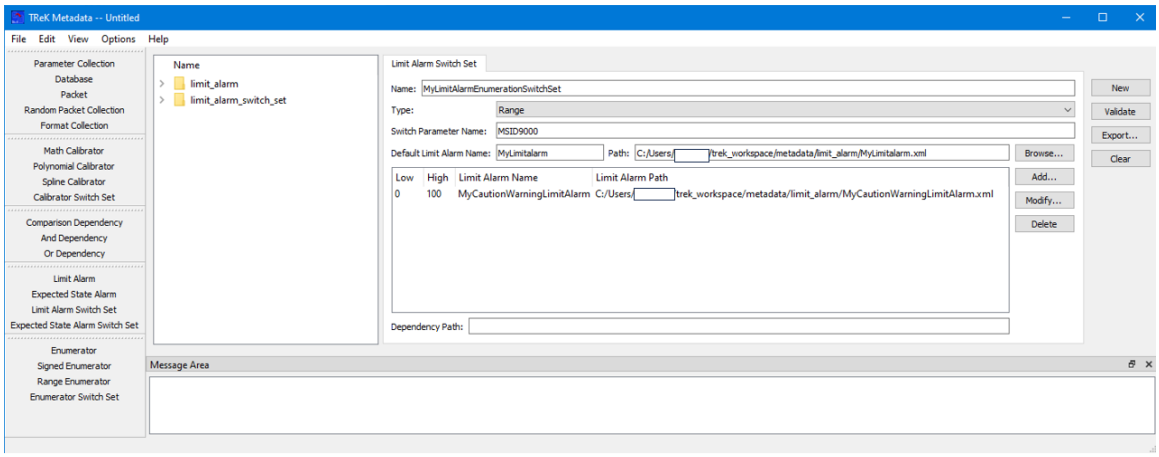


**Figure 40 Populated Limit Alarm Switch Set (Range)**

The following functions are available when working with a Limit Alarm Switch Set:

New
New provides the capability to clear the existing Limit Alarm Switch Set information.

Validate
Validate provides the capability to validate the Limit Alarm Switch Set information.

Export
Export provides the capability to save the Limit Alarm Switch Set to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.16 Expected State Alarm Switch Set

The Expected State Alarm Switch Set configuration provides the capability to work with an Expected State Switch Set. Figure 41 shows the Main Window in the Expected State Alarm Switch Set configuration.

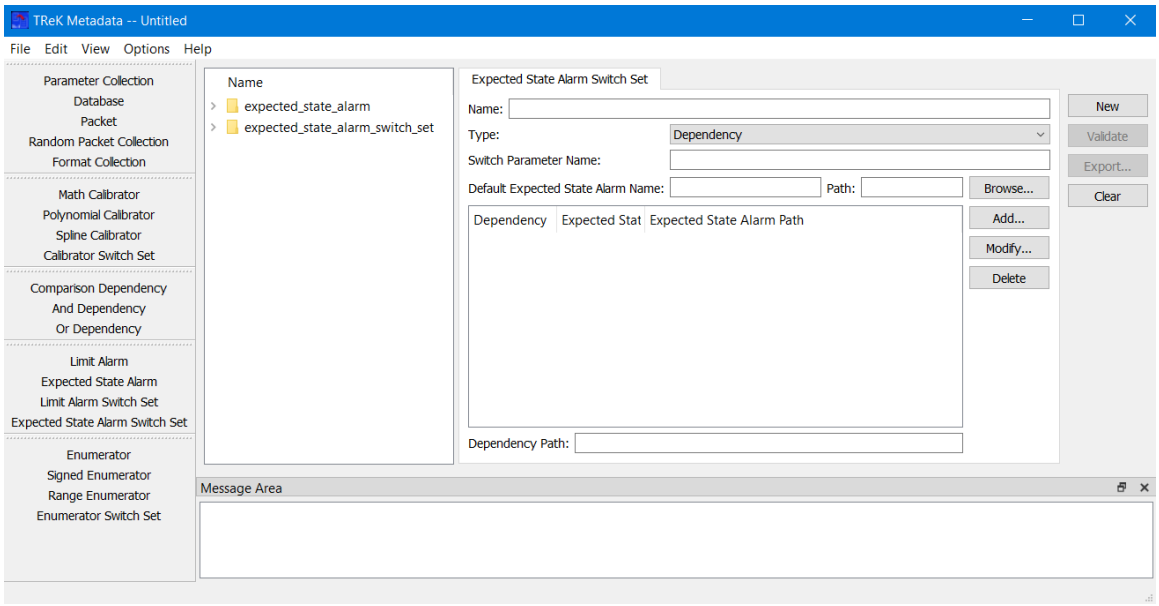


**Figure 41 Expected State Alarm Switch Set Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Expected State Alarms and Expected State Alarm Switch Sets in the trek_workspace. When you create an Expected State Alarm Switch Set and export it to the trek_workspace, it will be added to the Expected State Alarm Switch Set list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item outside of the trek_workspace, you will need to import the item. In the middle of the Main Window you will see the Work Area showing the Expected State Alarm Switch Set Tab. This is where you will define the details of the Expected State Alarm Switch Set. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Expected State Alarm Switch Set you are working on (e.g., Validate to validate the Expected State Alarm Switch Set or Export to export the Expected State Alarm Switch Set).

The Work Area provides the following information:

Name
The name field is used to display the name of the Expected State Alarm Switch Set.

Type
The type menu is used to specify the type of Expected State Alarm Switch Set – Dependency, Enumeration or Range.

Switch Parameter Name
The name of the parameter on which to switch.

Default Expected State Alarm Name
The default expected state alarm name field is used to display the name of the default expected state alarm. This must be the name of an expected state alarm that exists in the trek_workspace. This field is not editable. It will be populated when you enter a path into the Default Expected State Alarm Path field manually or by using the Browse button to select an Expected State Alarm.

Default Expected State Alarm Path
The default expected state alarm path field is used to display the path of the default expected state alarm. The default expected state alarm must exist in the trek_workspace. You can enter the path manually or you can use the Browse button to browse for an Expected State Alarm. If you delete the text in the Default Expected State Alarm Path field, the Default Expected State Alarm Name field will be cleared. If you enter information in the Default Expected State Alarm Path field that is invalid, the Default Expected State Alarm Name field will be cleared. If you enter valid information in the Default Expected State Alarm Path field manually or by using the Browse button the Default Expected State Alarm Name field will be automatically populated.

Set List
The Set List is used to enter set information for the Expected State Alarm Switch Set. The Set List area will be configured based on switch type. The Add button is used to add a row to the list. The Modify button is used to modify an existing item in the list. Modify will be available when an item in the list is selected. Enumeration and Range (Low and High) information can be edited using the Modify dialog or directly in the list. Dependency information is derived from the Dependency identified in the Modify dialog and cannot be edited in the list. The Delete button is used to delete a row from the list. When an Expected State Alarm is assigned to a set, the Expected State Alarm name and Expected State Alarm path will be displayed. When a Dependency row is selected, the Dependency path will be displayed in the Dependency Path field. Figure 42 shows a Dependency type switch. Figure 43 shows an Enumeration type switch. Figure 44 shows a Range type switch.

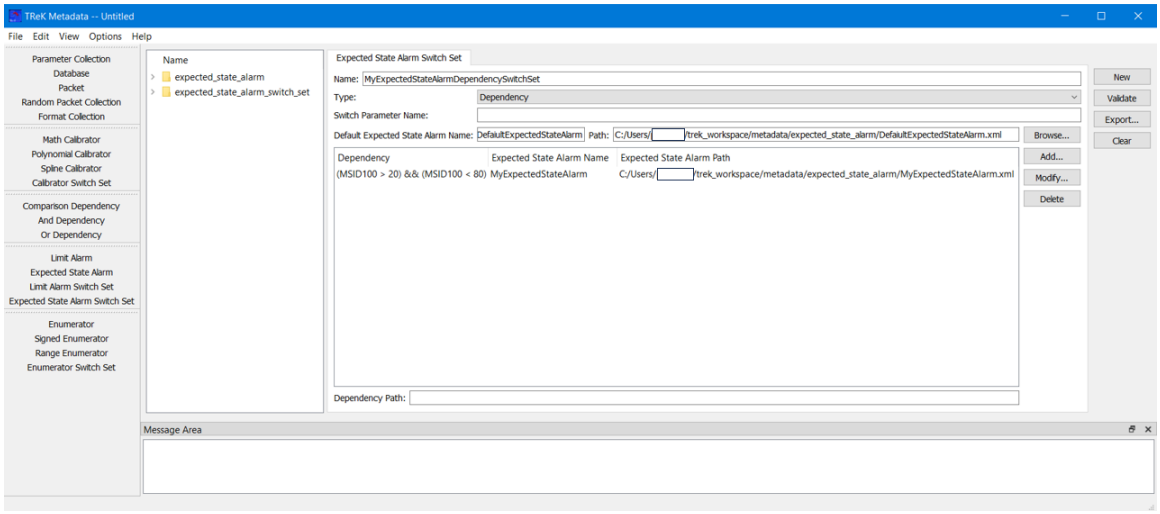


**Figure 42 Populated Expected State Alarm Switch Set (Dependency)**

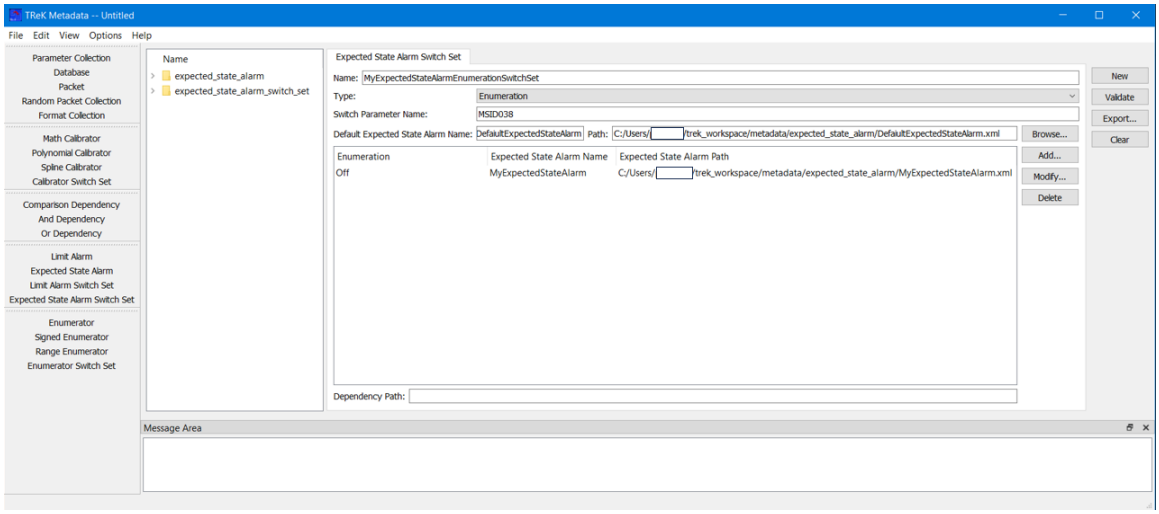


**Figure 43 Populated Expected State Alarm Switch Set (Enumeration)**

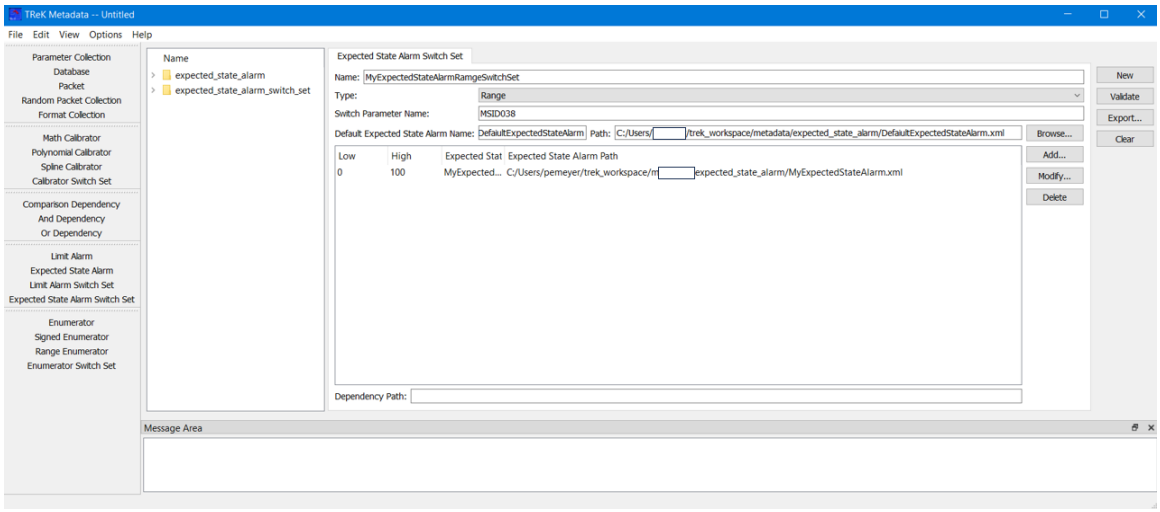


**Figure 44 Populated Expected State Alarm Switch Set (Range)**

The following functions are available when working with an Expected State Alarm Switch Set:

New
New provides the capability to clear the existing Expected State Alarm Switch Set information.

Validate
Validate provides the capability to validate the Expected State Alarm Switch Set information.

Export
Export provides the capability to save the Expected State Alarm Switch Set to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.17 Enumerator

The Enumerator configuration provides the capability to work with an Enumerator. Figure 45 shows the Main Window in the Enumerator configuration.

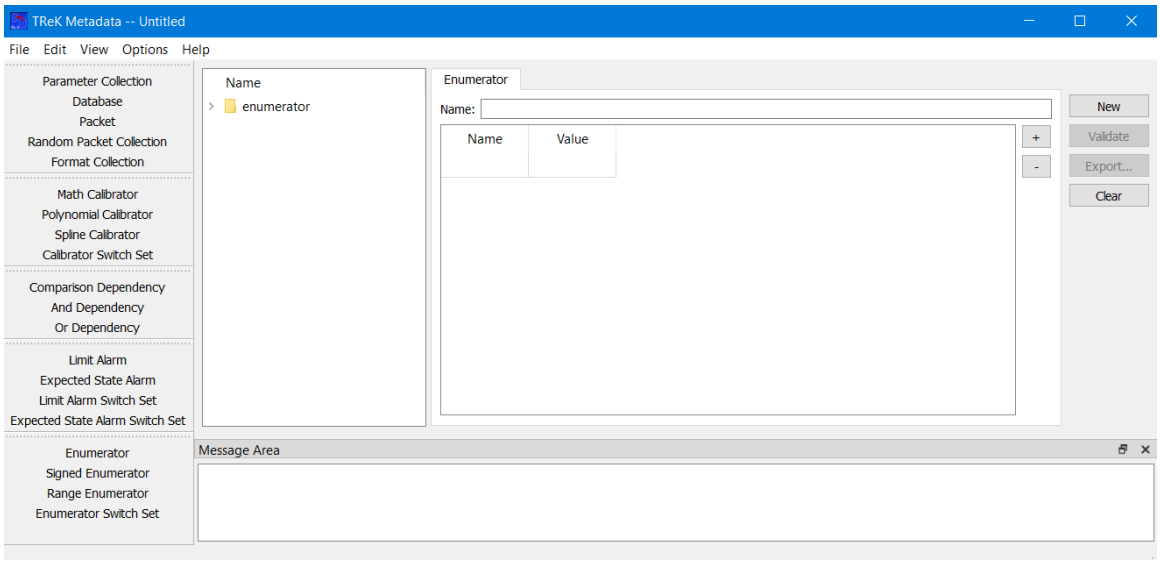


**Figure 45 Enumerator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Enumerators in the trek_workspace. When you create an Enumerator and export it to the trek_workspace, it will be added to the Enumerator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Enumerator Tab. This is where you will define the details of the Enumerator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Enumerator you are working on (e.g., Validate to validate the Enumerator or Export to export the Enumerator).

The Work Area provides the following information:

Name
The name field is used to display the name of the Enumerator.

Name Value List
The Name Value List is used to enter the Name and Value information for the Enumerator. The + (plus) button is used to add a row to the list and the – (minus) button is used to delete a row from the list. Name and Value information can be entered directly into the list as shown in Figure 46.

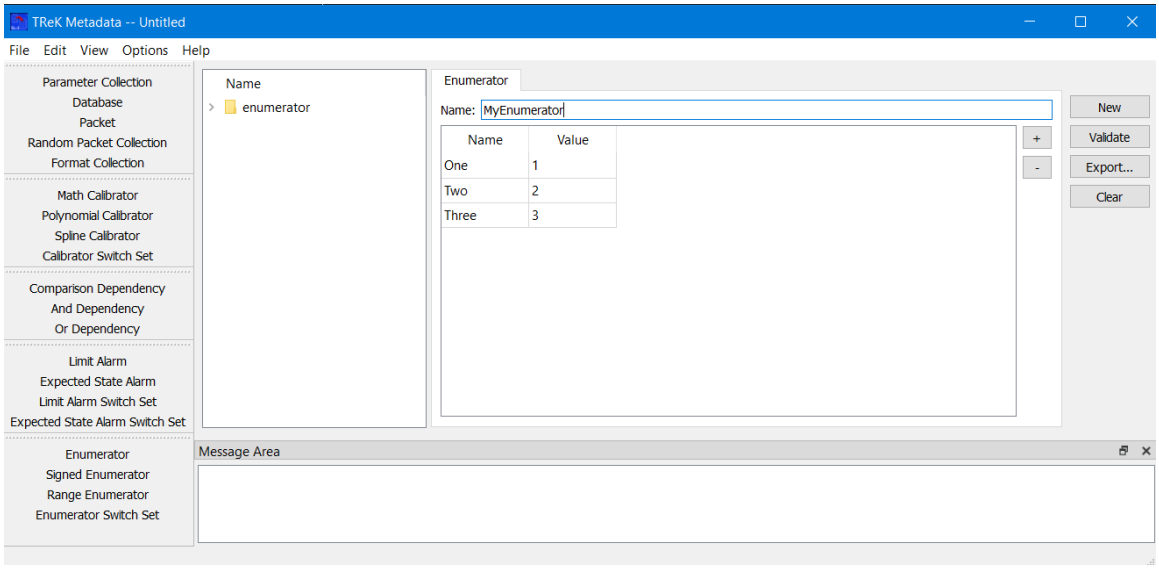


**Figure 46 Populated Enumerator**

The following functions are available when working with an Enumerator:

New
New provides the capability to clear the existing Enumerator information.

Validate
Validate provides the capability to validate the Enumerator information.

Export
Export provides the capability to save the Enumerator to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.18 Signed Enumerator

The Signed Enumerator configuration provides the capability to work with a Signed Enumerator. Figure 47 shows the Main Window in the Signed Enumerator configuration.

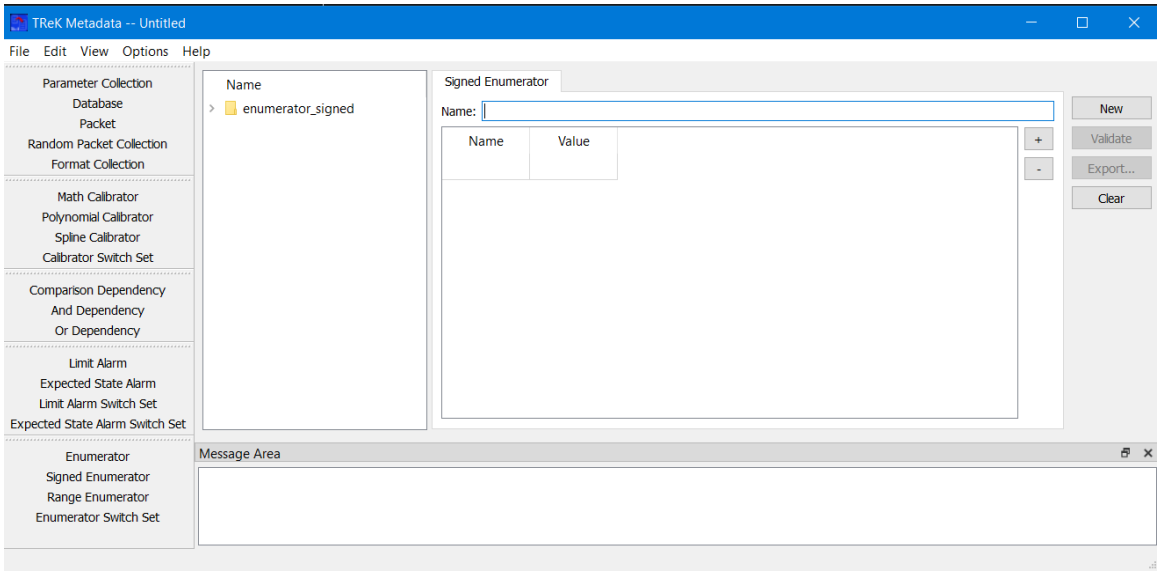


**Figure 47 Signed Enumerator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Signed Enumerators in the trek_workspace. When you create a Signed Enumerator and export it to the trek_workspace, it will be added to the Signed Enumerator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Signed Enumerator Tab. This is where you will define the details of the Signed Enumerator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Signed Enumerator you are working on (e.g., Validate to validate the Enumerator or Export to export the Enumerator).

The Work Area provides the following information:

Name
The name field is used to display the name of the Signed Enumerator.

Name Value List
The Name Value List is used to enter the Name and Value information for the Signed Enumerator. The + (plus) button is used to add a row to the list and the – (minus) button is used to delete a row from the list. Name and Value information can be entered directly into the list as shown in Figure 48.

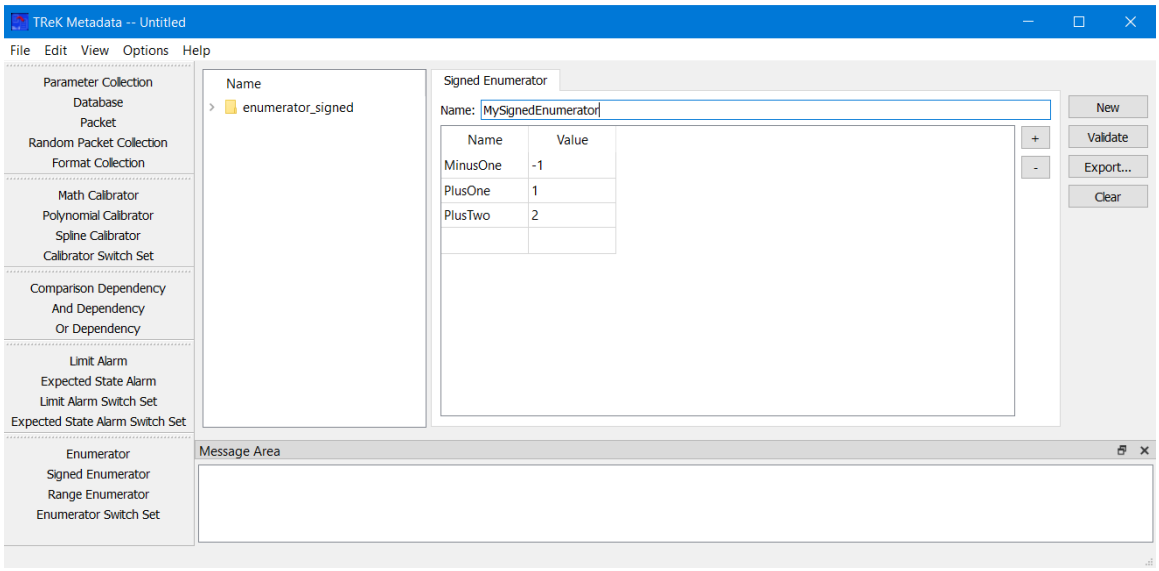


**Figure 48 Populated Signed Enumerator**

The following functions are available when working with a Signed Enumerator:

New
New provides the capability to clear the existing Signed Enumerator information.

Validate
Validate provides the capability to validate the Signed Enumerator information.

Export
Export provides the capability to save the Signed Enumerator to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.19 Range Enumerator

The Range Enumerator configuration provides the capability to work with a Range Enumerator. Figure 49 shows the Main Window in the Range Enumerator configuration.

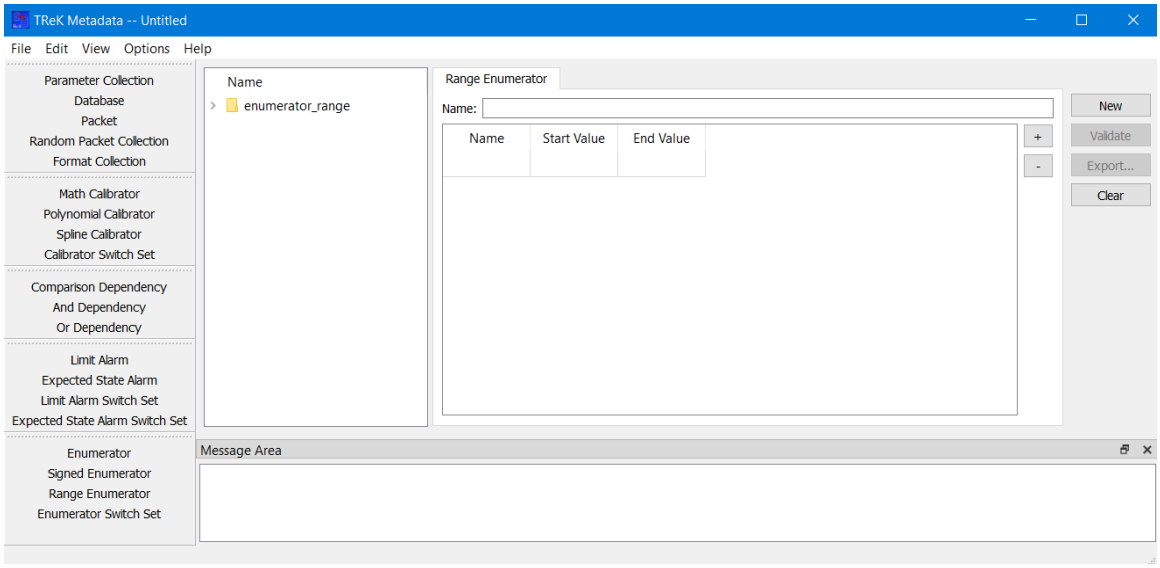


**Figure 49 Range Enumerator Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Range Enumerators in the trek_workspace. When you create a Range Enumerator and export it to the trek_workspace, it will be added to the Range Enumerator list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it into the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Range Enumerator Tab. This is where you will define the details of the Range Enumerator. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Range Enumerator you are working on (e.g., Validate to validate the Range Enumerator or Export to export the Range Enumerator).

The Work Area provides the following information:

Name

The name field is used to display the name of the Enumerator.

Value Range List

The Value Range List is used to enter the Name, Start Value, and End Value information for the Range Enumerator. The + (plus) button is used to add a row to the list and the – (minus) button is used to delete a row from the list. Name, Start Value, and End Value information can be entered directly into the list as shown in Figure 50.

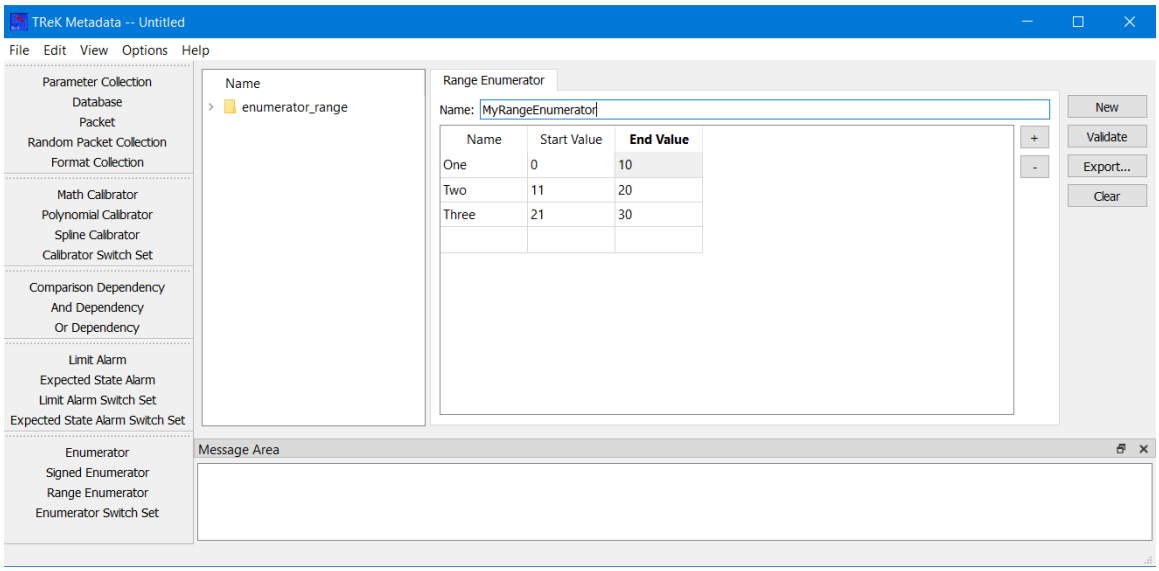


**Figure 50 Populated Range Enumerator**

The following functions are available when working with a Range Enumerator:

New
New provides the capability to clear the existing Range Enumerator information.

Validate
Validate provides the capability to validate the Range Enumerator information.

Export
Export provides the capability to save the Range Enumerator to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.20 Enumerator Switch Set

The Enumerator Switch Set configuration provides the capability to work with an Enumerator Switch Set. Figure 51 shows the Main Window in the Enumerator Switch Set configuration.

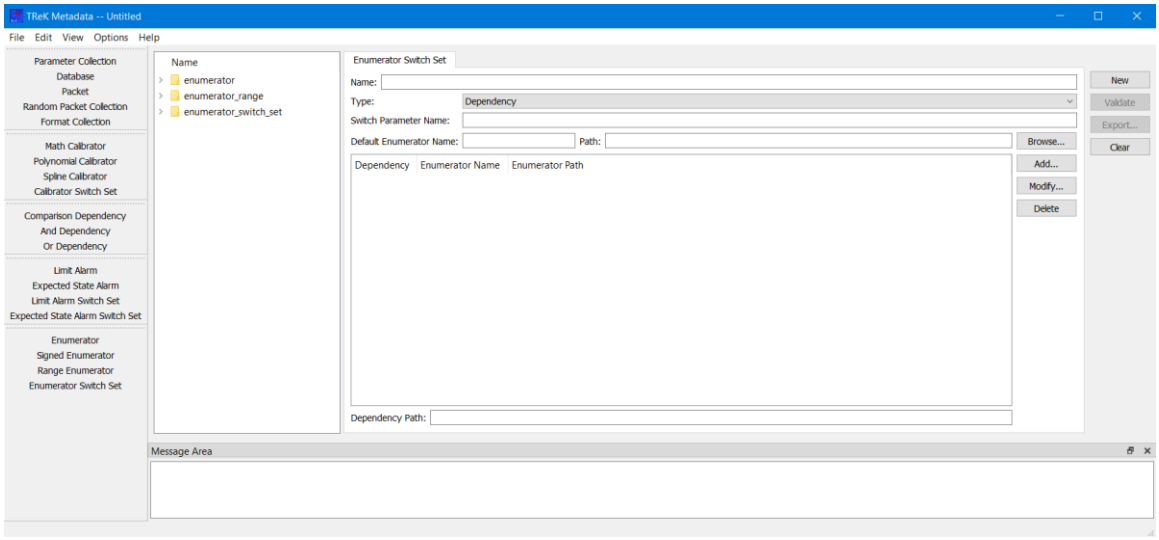


**Figure 51 Enumerator Switch Set Configuration**

On the left-hand side of the Main Window you will see the Library Area. It is configured to provide access to existing Enumerators and Enumerator Switch Sets in the trek_workspace. When you create an Enumerator Switch Set and export it to the trek_workspace, it will be added to the Enumerator Switch Set list. The Library Area only displays items stored in the trek_workspace. If you need to work on an item that is outside of the trek_workspace, you will need to move it to the trek_workspace. In the middle of the Main Window you will see the Work Area showing the Enumerator Switch Set Tab. This is where you will define the details of the Enumerator Switch Set. On the right-hand side of the Main Window you will see a set of buttons that provide various capabilities applicable to the Enumerator Switch Set you are working on (e.g., Validate to validate the Enumerator Switch Set or Export to export the Enumerator Switch Set).

The Work Area provides the following information:

Name
The name field is used to display the name of the Enumerator Switch Set.

Type
The type menu is used to specify the type of Enumerator Switch Set – Dependency, Enumeration or Range.

Switch Parameter Name
The name of the Parameter on which to switch.

Default Enumerator Name
The default enumerator name field is used to display the name of the default enumerator. This must be the name of an enumerator that exists in the trek_workspace. This field is not editable. It will be populated when you enter a path into the Default Enumerator Path field manually or by using the Browse button to select an Enumerator.

Default Enumerator Path

The default enumerator path field is used to display the path of the default enumerator. The default enumerator must exist in the trek_workspace. You can enter the path manually or you can use the Browse button to browse for an Enumerator. If you delete the text in the Default Enumerator Path field, the Default Enumerator Name field will be cleared. If you enter information in the Default Enumerator Path field that is invalid, the Default Enumerator Name field will be cleared. If you enter valid information in the Default Enumerator Path field manually or by using the Browse button the Default Enumerator Name field will be automatically populated.

Set List

The Set List is used to enter set information for the Enumerator Switch Set. The Set List area will be configured based on switch type. The Add button is used to add a row to the list. The Modify button is used to modify an existing item in the list. Modify will be available when an item in the list is selected. Enumeration and Range (Low and High) information can be edited using the Modify dialog or directly in the list. Dependency information is derived from the Dependency identified in the Modify dialog and cannot be edited in the list. The Delete button is used to delete a row from the list. When an Enumerator is assigned to a set, the Enumerator name and Enumerator path will be displayed. When a Dependency row is selected, the Dependency path will be displayed in the Dependency Path field. Figure 52 shows a Dependency type switch. Figure 53 shows an Enumeration type switch. Figure 54 shows a Range type switch.

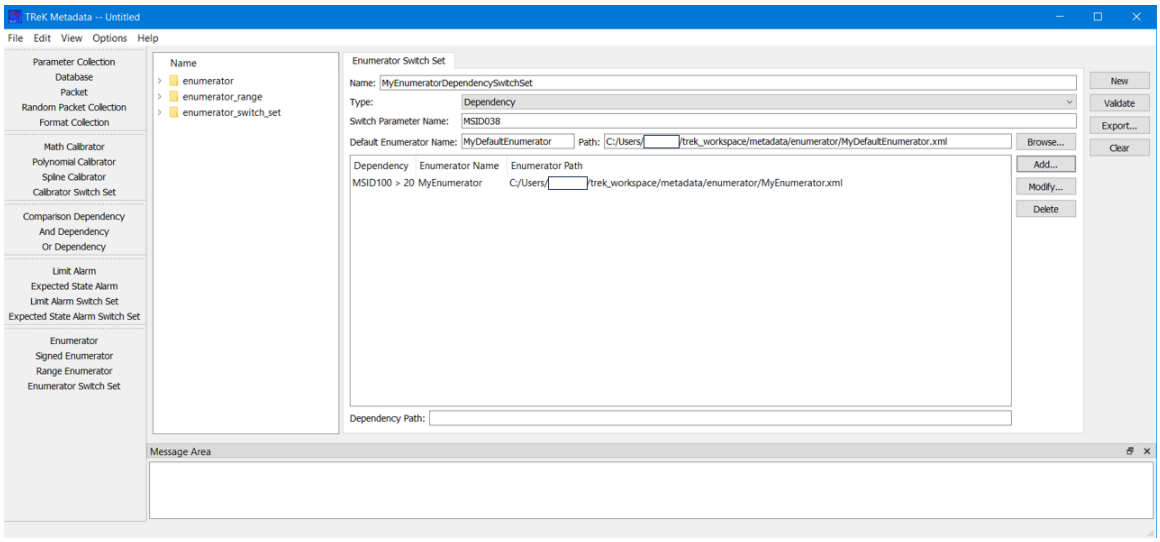


**Figure 52 Populated Enumerator Switch Set (Dependency)**

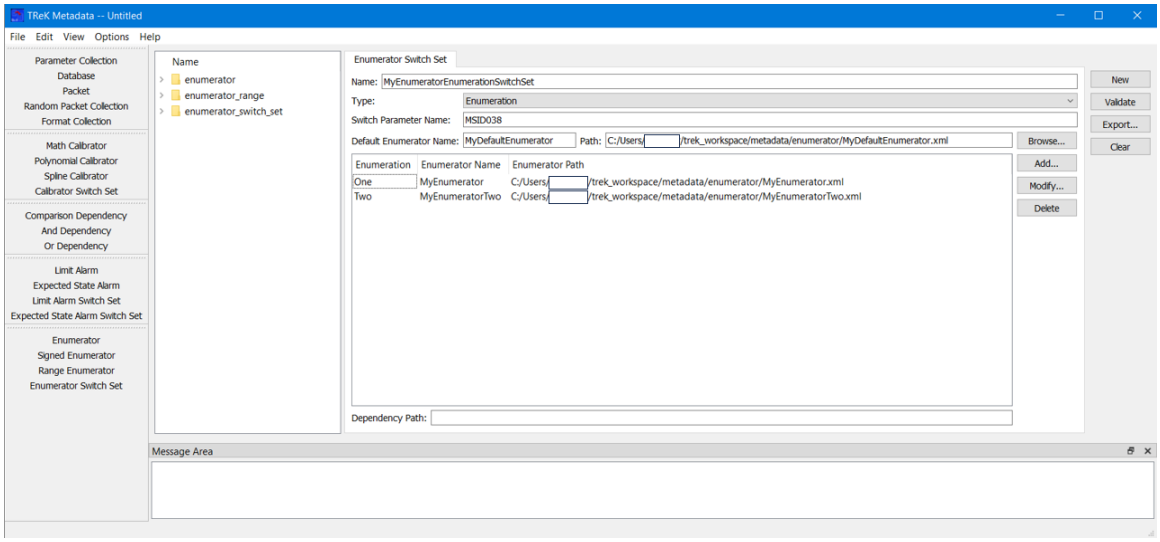


**Figure 53 Populated Enumerator Switch Set (Enumeration)**

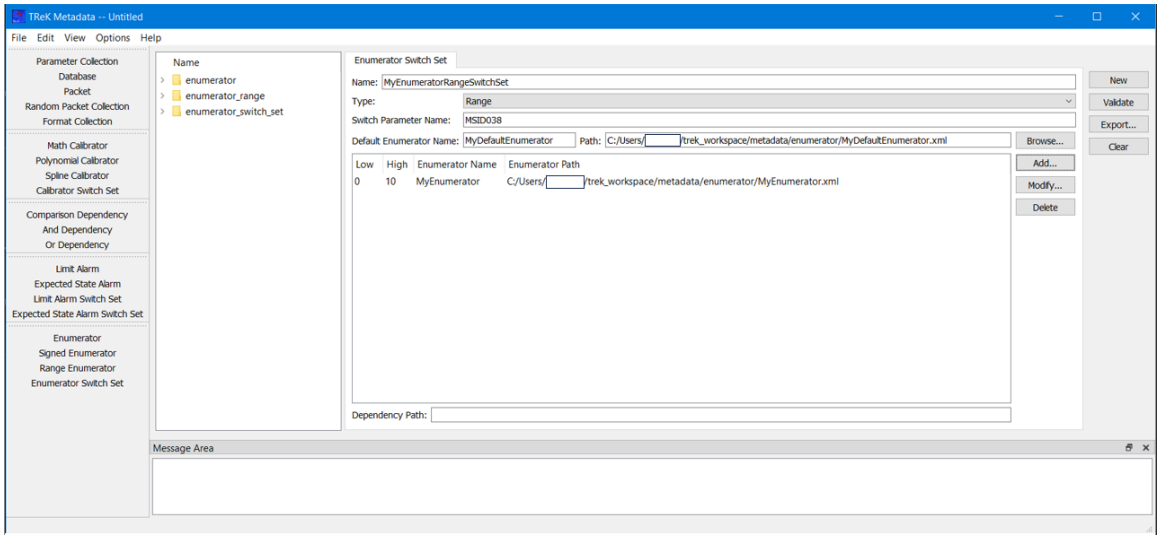


**Figure 54 Populated Enumerator Switch Set (Range)**

The following functions are available when working with an Enumerator Switch Set:

New
New provides the capability to clear the existing Enumerator Switch Set information.

Validate
Validate provides the capability to validate the Enumerator Switch Set information.

Export
Export provides the capability to save the Enumerator Switch Set to a file.

Clear
Clear provides the capability to clear the Work Area.

## 6.21 Export

The Export dialog is shown in Figure 55. This dialog is context sensitive and will show the Export Options available based on the user's choice from the toolbar. Figure 55 shows an export dialog from the Parameter Collection selection.

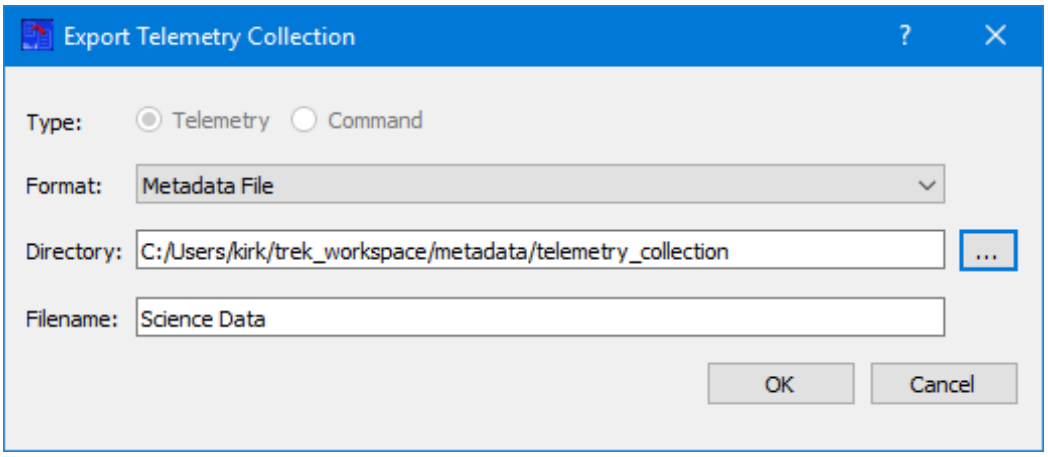


**Figure 55 Export Dialog for Parameter Collection**

Each field is described below:

Format
The Format menu provides the capability to select the format of the file.

Directory
The Directory field provides the capability to enter the Directory location for the exported element. It defaults to the appropriate directory in the trek_workspace.

Filename
The Filename field provides the capability to enter the filename for the exported item. It will default to the text entered in the Name field in the Work Area.

## 6.22 Import

The Import dialog is shown in Figure 56. This dialog is context sensitive and will show the Import Options available based on the configuration of the application.

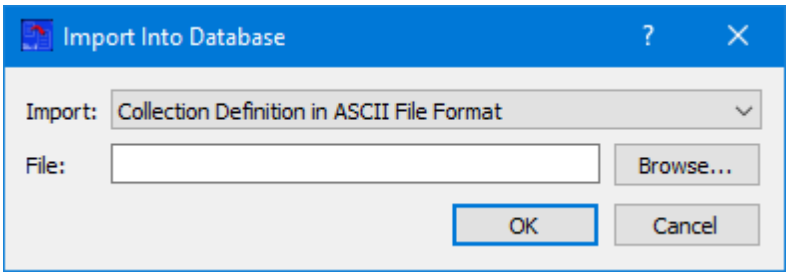


**Figure 56 Import Dialog**

Each field is described below:

Format
The Format menu provides the capability to identify the format of the file. If the file does not match the format identified, the import will fail.

File
The File field provides the capability to enter the absolute path for the item to import.

## 6.23 Parameter Details

The Parameter Details dialog is shown in Figure 57. The Parameter Details dialog provides access to the properties of the selected parameter. For information about the items in this dialog please reference the TReK Data library parameter class definition.

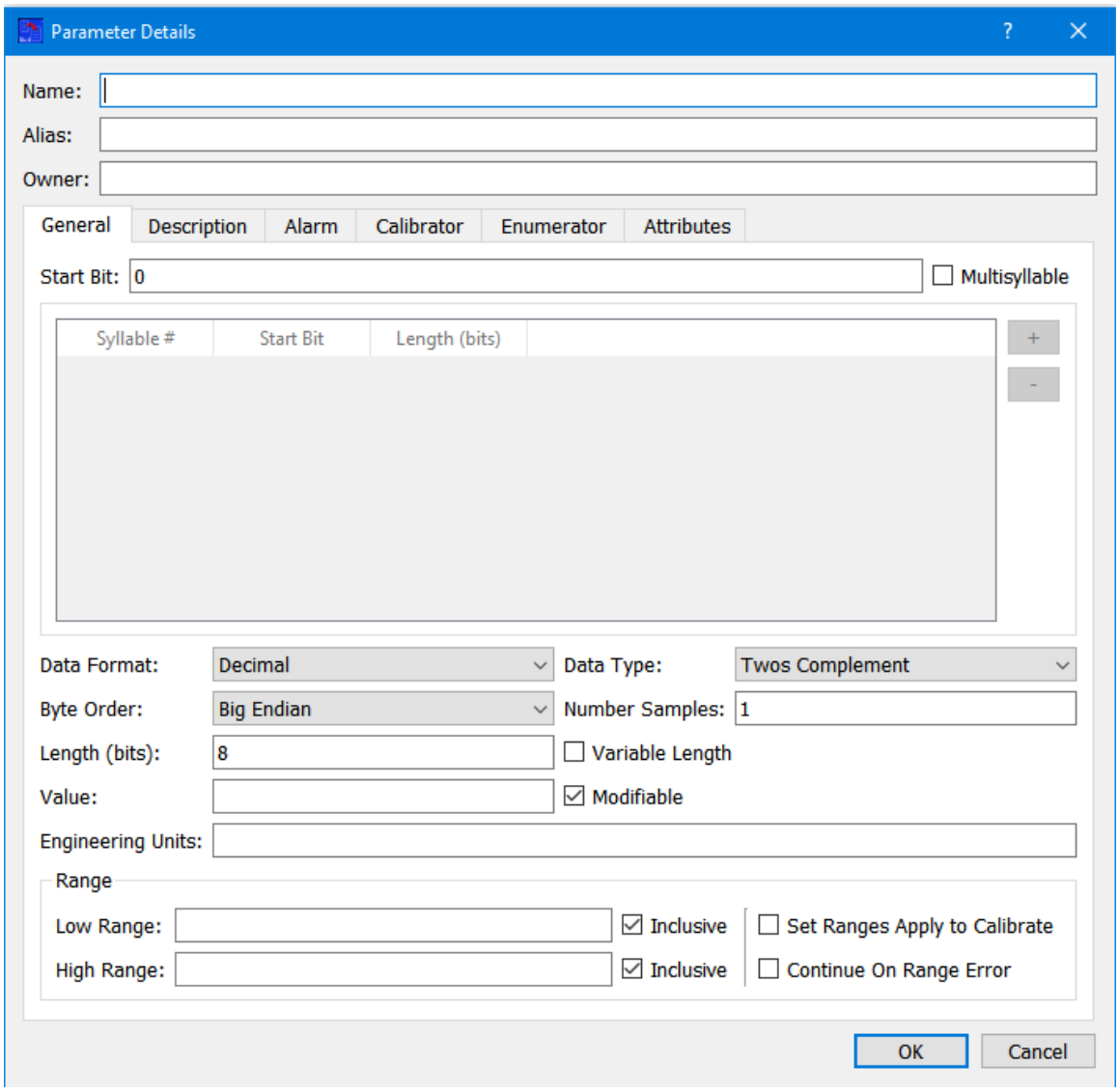


**Figure 57 Parameter Details Dialog**

Note: When the Multisyllable checkbox is checked, the Start Bit field is not used, and the field is set to read-only. Syllable numbers start at 1. The total of the syllable lengths must equal the overall parameter length.

## 6.24 Packet Properties

The Packet Properties dialog is shown in Figure 58. The Packet Properties dialog provides the capability to define one or more properties in a packet. For details on Packet Properties please reference the TReK Concepts document. Certain tabs contain a checkbox to indicate if the Packet Properties should be set in the Packet. If the checkbox is set, all the data entered on the tab will be validated and the attribute will be added to the Packet.

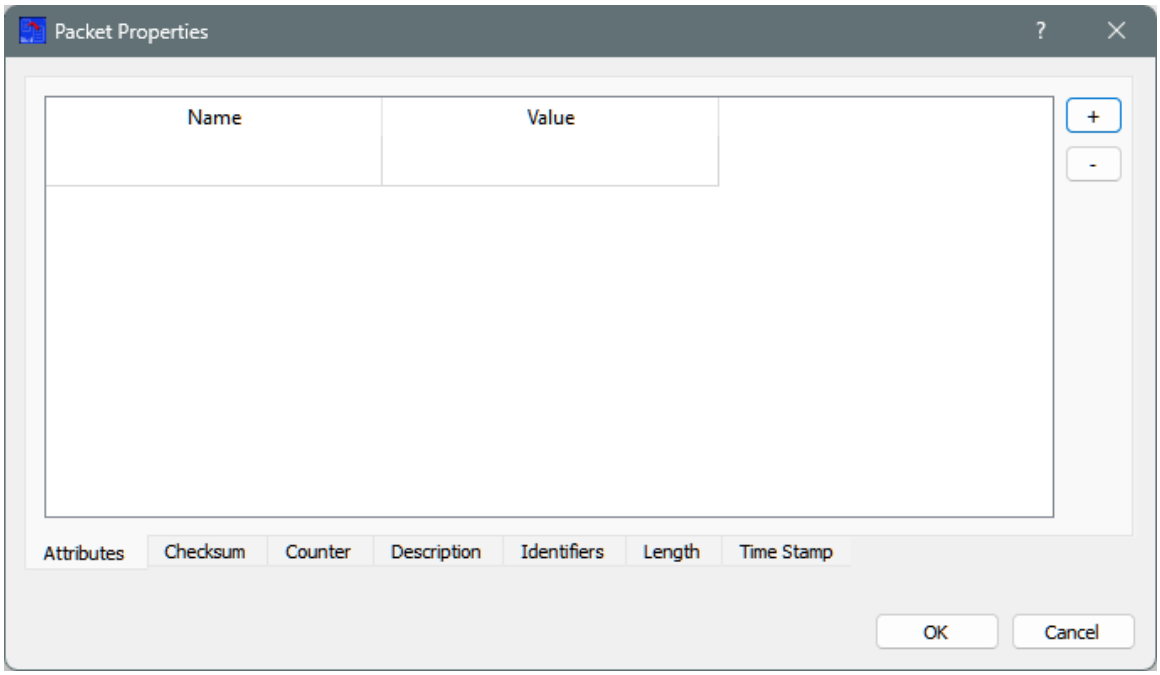


**Figure 58 Packet Properties Dialog**

## 6.25 Add EHS PDSS Payload Pass-Thru Packet to TReK Database

The Add EHS PDSS Payload Pass-Thru Packet to TReK Database dialog is shown in Figure 59. This capability is available from the Options menu and is used to add an EHS PDSS Payload pass-thru packet to a TReK database. You can choose to add the pass-thru packet to an existing database or a new database. If your packet is defined as a pass-thru packet, you can use the TReK API to retrieve the header parameters or the data zone portion of the packet, but not any of the individual parameters in the data zone of the packet. Once you have added a pass-thru packet to a TReK Database, you can use capabilities on the Metadata application's Database tab to export the packet definition to a TReK metadata file.

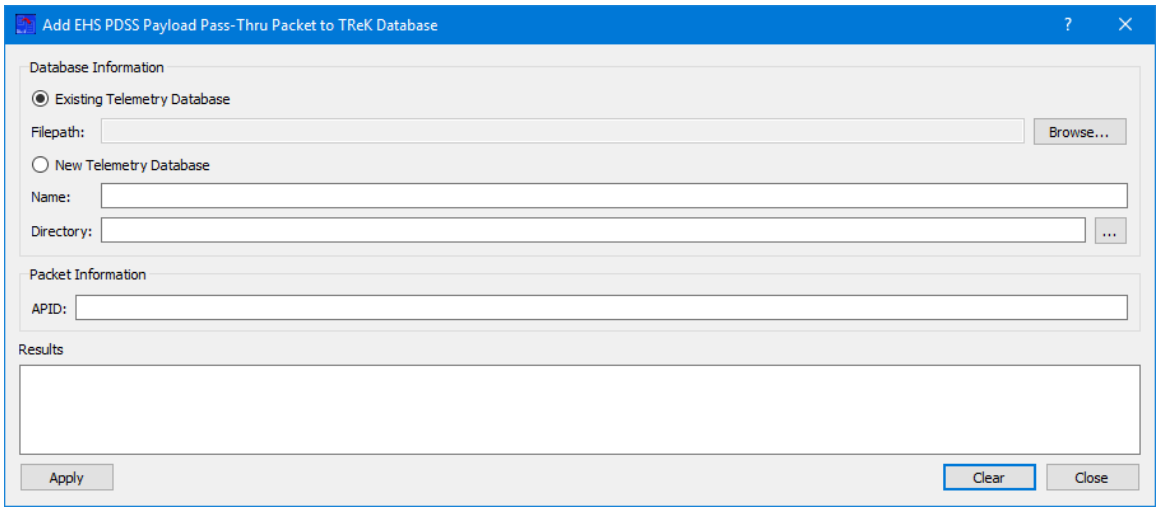


**Figure 59 Add EHS PDSS Payload Pass-Thru Packet to TReK Database Dialog**

Each field is described below:

Filepath
This is the absolute path to an existing TReK telemetry database.

Name
This is the absolute path for the directory to use for the new TReK telemetry database.

Directory
This is the name for the new TReK telemetry database.

APID
This is the APID for the pass-thru packet.

Results
The Results area displays messages about the results of adding the pass-thru packet to the database. Any information, warning, or error messages will be displayed here.

Apply
The Apply button is used to add the pass-thru packet to the database. Any information, warning, or error messages will be displayed in the Results area.

Clear
The Clear button clears all the fields in the dialog.

Close
The Close button closes the dialog.

## 6.26 Convert EHS Partial Database Files Into TReK Database

The Convert EHS Partial Database Files Into TReK Database dialog is shown in Figure 60. This capability is available from the Options menu and is used to import data in EHS partial database files into a new TReK database. The new TReK database is created during the convert. The convert process can take a while if the EHS partial database files contain a large amount of data. You can continue to work on other tasks in the application while the convert is in work.

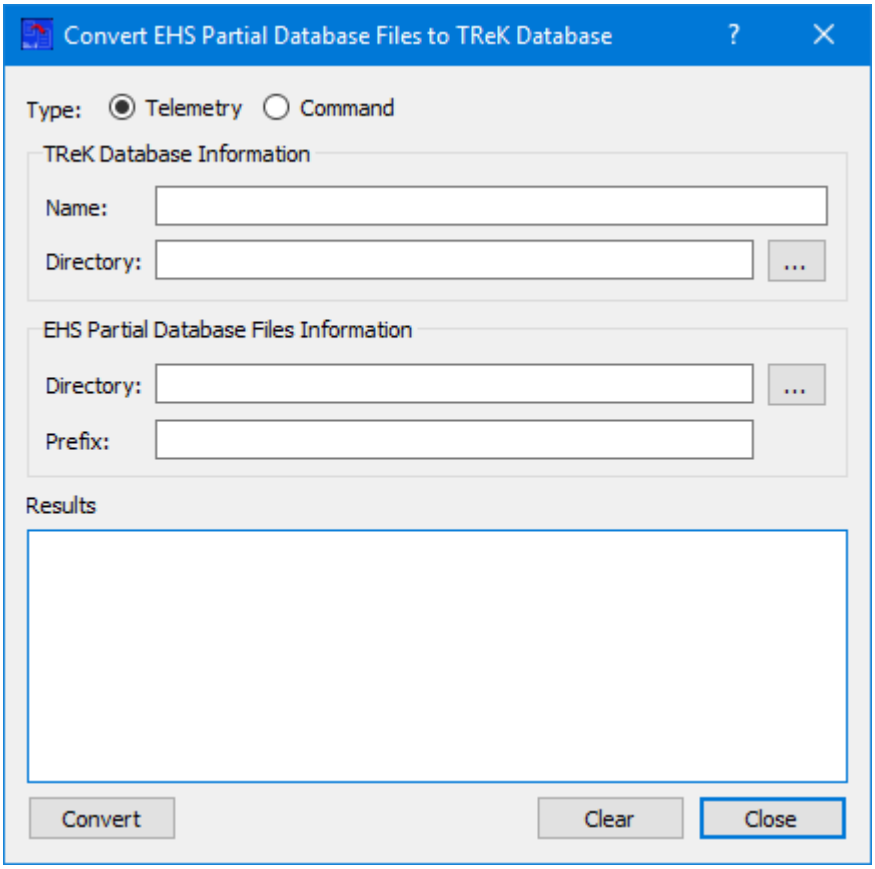


**Figure 60 Convert EHS Partial Database Files Into TReK Database Dialog**

Each field is described below:

Type
The Type is used to identify whether Telemetry information or Command information is to be imported.

Name
The Name field is used to enter the name of the new TReK database that will be created. If the Telemetry radio button is selected, a new TReK Telemetry database will be created.

If the Command radio button is selected, a new TReK Command database will be created.

Directory
The Directory field is used to enter the directory where the EHS partial database download files reside. All the files for the convert (import) should be in one directory.

Prefix
Filenames for EHS Partial Database files with telemetry information contain a prefix number. This number should be entered in the Prefix field. This field is not applicable for command information and will be grayed out when the Command radio button is selected.

Results
The Results area is used to display messages about the results of converting the data from the EHS partial database files to import it into the new TReK database. Any information, warnings, or errors will be displayed here.

Convert
The Convert button is used to execute the convert (import). Messages related to the convert process will be displayed in the Results area. The process can take a while if the EHS partial database files contain a large amount of data. You can continue to work on other tasks in the application while the convert is in work.

Clear
The Clear button clears the Results area.

Close
The Close button closes the dialog.

### 6.27 Convert TReK Database Format

The Convert TReK Database Format dialog is shown in Figure 61 and Figure 62. This capability is available from the Options menu and is used to convert a TReK database from a previous database format to a new database format. A new TReK database is created during the convert. The convert process can take a while if the database file(s) contain a large amount of data. You can continue to work on other tasks in the application while the convert is in work. There are two options available. You can convert database files that reside in the TReK workspace as shown in Figure 61 or you can convert database files that reside outside the TReK workspace as shown in Figure 62.

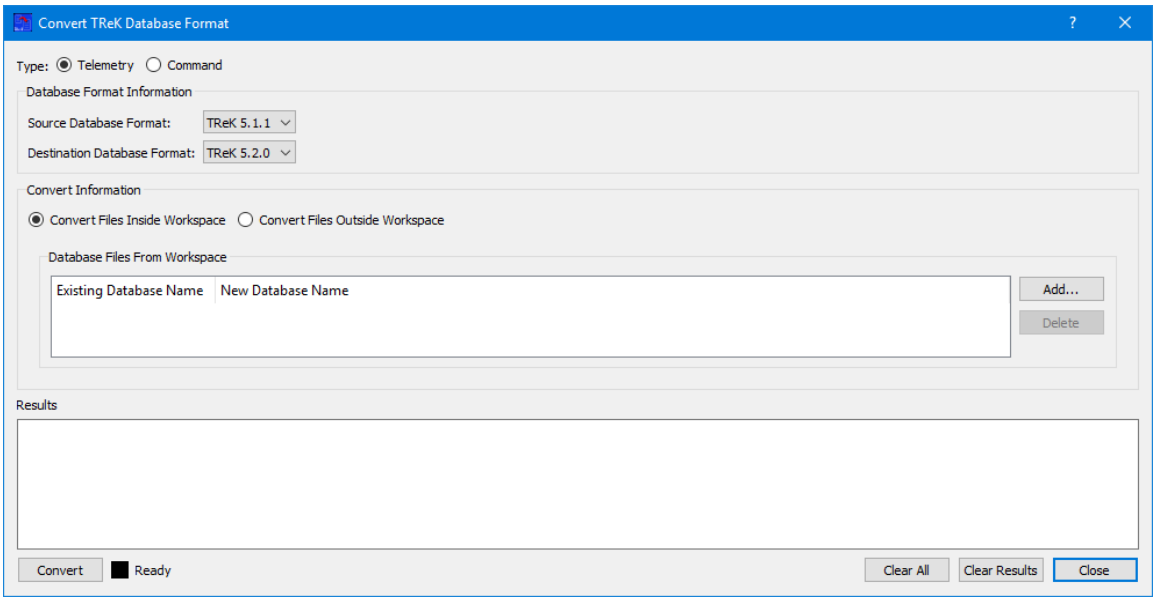


**Figure 61 Convert TReK Database Format (Inside Workspace)**

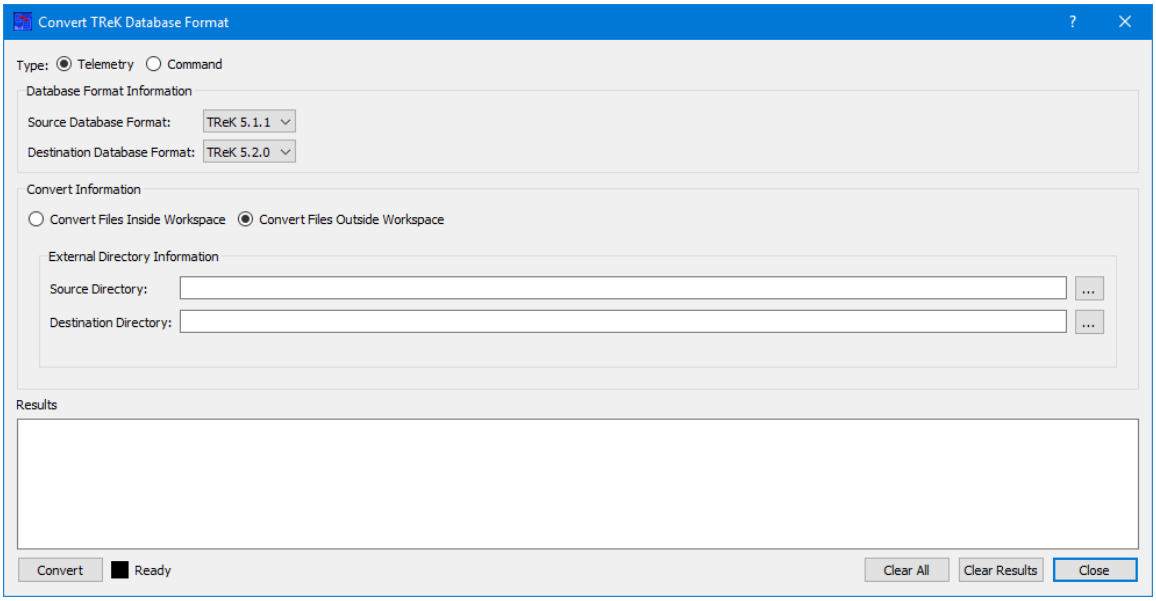


**Figure 62 Convert TReK Database Format (Outside Workspace)**

Each field is described below:

Type

The Type identifies whether a Telemetry database or Command database is to be converted.

Source Database Format

Identifies the format of the database to convert.

Destination Database Format
Identifies the format for the new database.

Convert Files Inside Workspace
Specifies to convert files that exist inside the TReK workspace.

Convert Files Outside Workspace
Specifies to convert files that exist outside the TReK workspace.

Database Files From Workspace
This list identifies existing files in the TReK workspace to be converted. The **Existing Database Name** column will display the name of the file that is added to the list. The corresponding **New Database Name** column will display the existing filename with the new format information appended to the end of the name. For example, if the existing filename is TelemetyDatabase.db, the new default filename will be TelemetryDatabase520.db. The new filename can be modified by double clicking in the cell and editing the name.

Add
The Add button is used to browse the TReK workspace and select existing database files to convert. The Add will only add existing files located in a TReK workspace metadata database directory that have a .db extension. The Add will not add duplicate items to the list.

Delete
The Delete button is used to delete selected items in the list.

Source Directory
The Source directory is a directory outside the TReK workspace that contains one or more TReK databases to convert. The **…** button can be used to browse and select a directory.

Destination Directory
The Destination directory is a directory outside the TReK workspace where the new databases will be created by the convert. The **…** button can be used to browse and select a directory. The Destination directory must empty before the convert can be performed.

Results
The Results area is used to display messages about the results of the convert process. Any information, warnings, or errors will be displayed here.

Convert
The Convert button is used to perform the convert. Messages related to the convert process will be displayed in the Results area. The process can take a while if the database

files contain a large amount of data. You can continue to work on other tasks in the application while the convert is in work.

Clear All
The Clear All button clears all the fields in the dialog (this includes all the fields in the Database Files From the Workspace and External Directory Information areas).

Clear Results
The Clear Results button only clears the Results area.

Close
The Close button closes the dialog.

### 6.28 Export Metadata to CDEF

The Export Metadata to CDEF dialog is shown in Figure 63. This capability is available from the Options menu and is used to export one or more telemetry or command packets into a Common Data Exchange Format (CDEF) file.

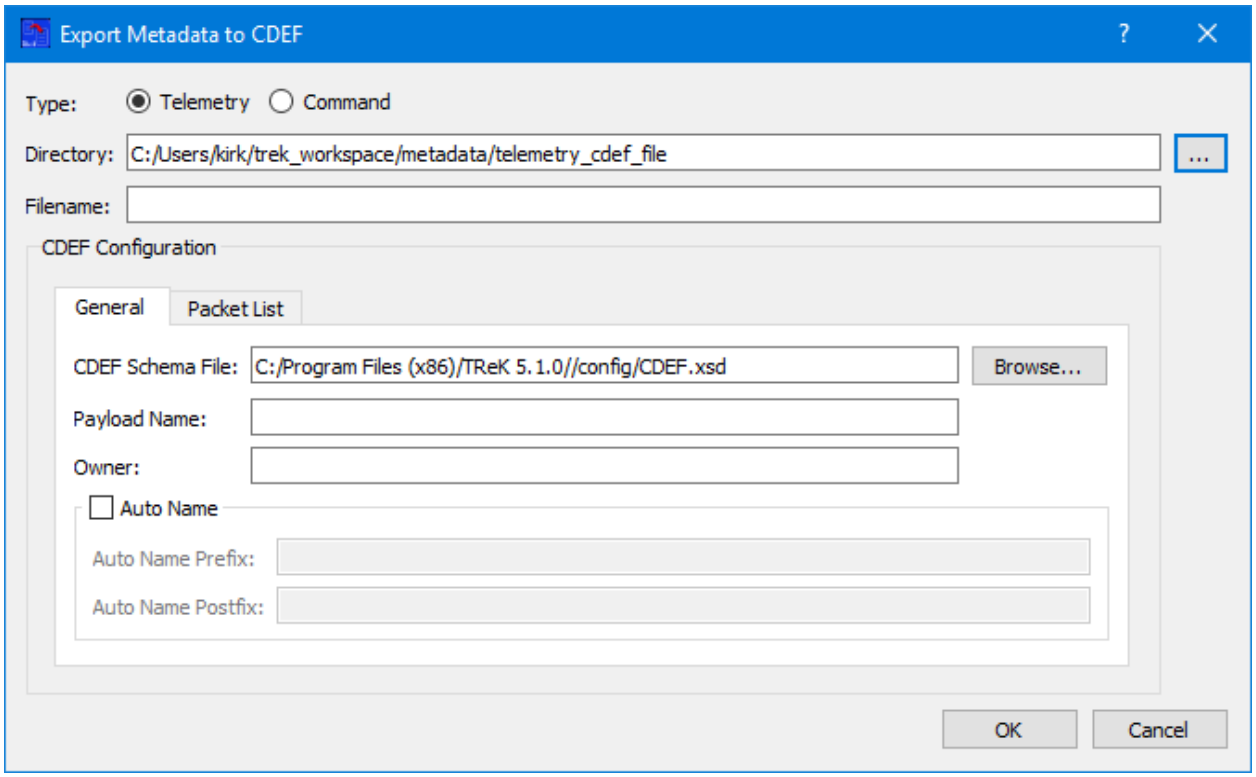


**Figure 63 Export Metadata to CDEF Dialog**

Each field is described below.

Directory
The Directory field provides the capability to enter the Directory location for the exported element. It defaults to the appropriate directory in the trek_workspace.

Filename
The Filename field provides the capability to enter the filename for the exported item.

The CDEF Configuration area provides the capability to enter CDEF specific information. Most of the information collected is to allow you to easily enter information that is required in the CDEF file but is not necessarily required in the internal TReK files and database. Each field is described below.

### 6.28.1 General Tab

The General tab is used to enter general information about the CDEF export.

CDEF Schema Location
This field is used to identify the location of the CDEF Schema. The Browse button can be used to browse the local file system for the file location.

Payload Name
This field is used to identify the Payload Name assigned by the ISS Program. If the data being exported was previously imported from CDEF, this field may be left blank.

Owner
This field is used to identify the Owner for the telemetry data. This is only used when an owner is not available for an individual item (e.g., parameter). If one already exists, it will not be overwritten. If the data being exported was previously imported from CDEF, this field may be left blank.

Auto Name Checkbox
Check the Auto Name Checkbox if auto naming should be used. If this is checked, the Auto Name Prefix and Auto Name Postfix are required. The ISS Program assigns an eight character name to payloads. If you did not use that name when creating you metadata for TReK, you can use this feature to have those names automatically generated. The names you provided for the parameters will be set to the ‘alias’ for the parameter to preserve more meaningful names.

Auto Name Prefix
This field is used to enter the Auto Name Prefix information.

Auto Name Postfix
This field is used to enter the Auto Name Postfix information. The postfix is a single character (typically ‘U’) that is added as the last character of the parameter name.

### 6.28.2 Packet List Tab

The Packet List tab is used to identify one or more packets to be included in the CDEF export. It is shown in Figure 64. The Browse Button is used to browse a TReK Database to select and add a packet to the list. The Browse Disk button is used to browse the local file system to select and add a packet using a TReK metadata file definition. The Delete button is used to delete an item from the list. If the export is for telemetry, all payload telemetry metadata information to be shared across systems (ISS, POIC, TReK etc.) must be included in the CDEF file. Many payload teams will only need to include health and status metadata. If the export is for command, all payload command metadata information to be shared across systems (ISS, POIC, TReK etc.) must be included in the CDEF file.

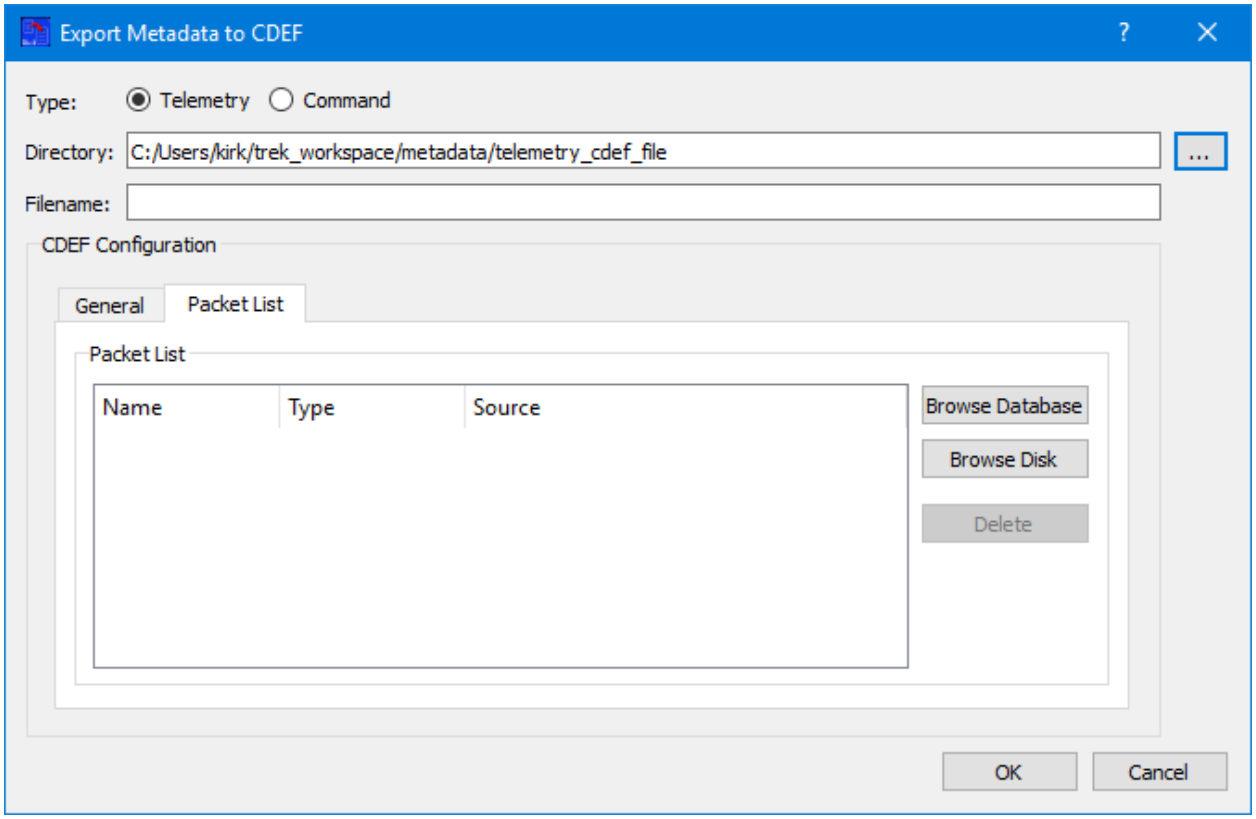


**Figure 64 Export Metadata to CDEF (Packet List Tab)**

## 6.29 Import Metadata from CDEF

The Import Metadata to CDEF dialog is shown in Figure 65 Import Metadata from CDEF. This capability is available from the Options menu and is used to import one or more telemetry or command packets from a Common Data Exchange Format (CDEF) file.

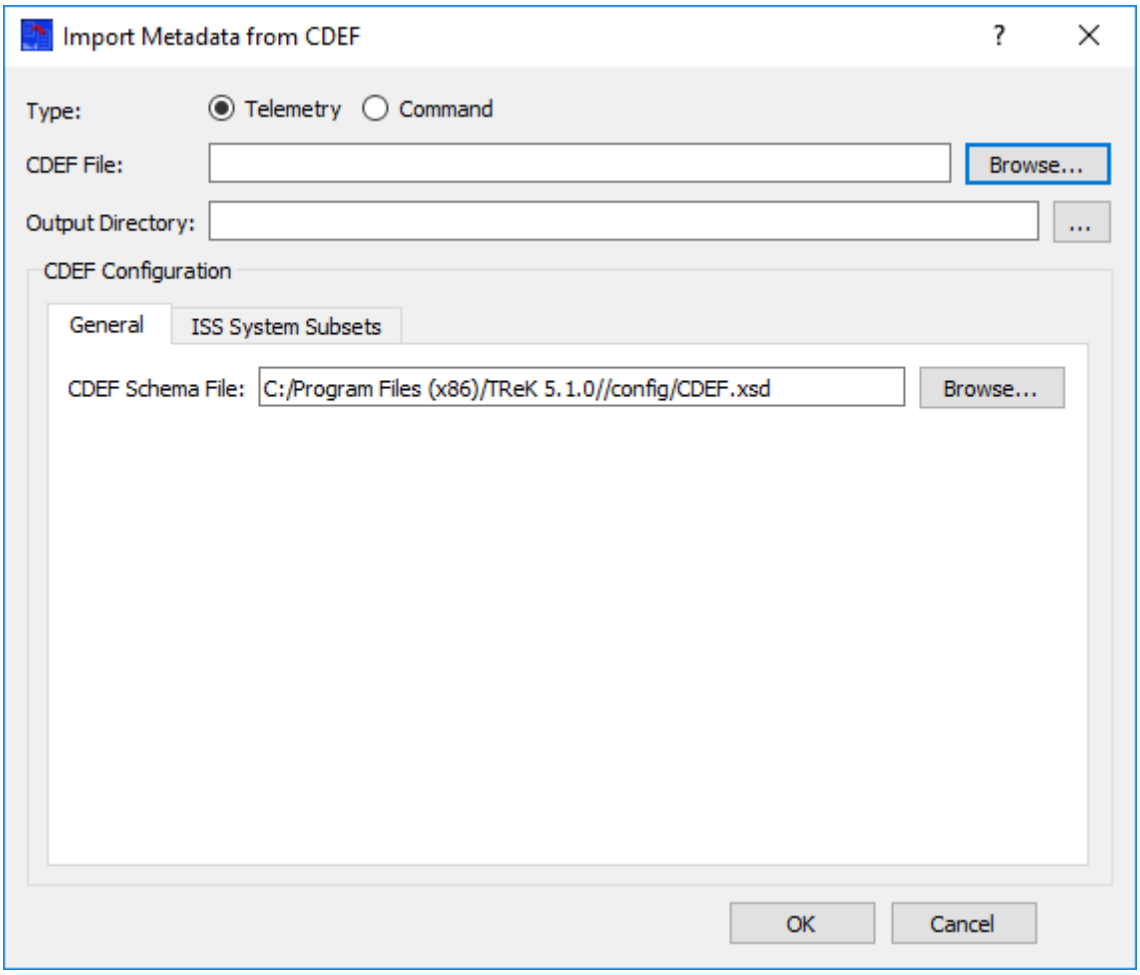


**Figure 65 Import Metadata from CDEF**

Each field is described below.

CDEF File
The CDEF File to import.

Output Directory
The Directory field provides the capability to enter the Directory location for the TReK metadata files generated during the import. It defaults to the appropriate directory in the trek_workspace. The metadata filenames will contain the packet type and APID for telemetry data and the command name for command data.

*Note: If you select a directory different than the default for exporting telemetry metadata, some information will still be written in the trek_workspace to allow for editing of the lowest level information. Files will be created in the telemetry_collection, telemetry_packet, and random_packet_collection subdirectories.*

The CDEF Configuration area provides the capability to enter CDEF configuration information. Most of the information collected is to allow you to easily enter information that is required to perform the CDEF Import. Each field is described below.

### 6.29.1 General Tab

The General tab is used to enter general information about the CDEF import.

CDEF Schema Location
This field is used to identify the location of the CDEF Schema. The Browse button can be used to browse the local file system for the file location.

### 6.29.2 ISS System Subsets Tab

The ISS Systems Subsets tab is used to add ISS system subsets to APIDs in the CDEF Import to allow processing of health and status data during standalone testing and/or flight. It is shown in Figure 66. There are two options to add the subsets to the imported metadata.

The default option adds the user specified subsets in addition to any subsets provided in the CDEF file. The + (plus) button is used to add a row to the list. The – (minus) button is used to delete a row from the list. The Subset information can be entered directly into the list. The list should default to the ISS system subsets needed, but the lengths may change if the ISS system software is updated. The default lengths are those known at the time of this TReK release.

The second option reads the information about subsets from EHS partial database files. These are the same files used in Section 6.26, but does not require converting the files into a TReK telemetry database. Subsets defined in the CDEF file will replace any data from the EHS partial database files. Only minimal information about other subsets is added to the resulting metadata file.

Since the number of subsets for flight is very large and changes at least a couple of times a year, this option is best for importing CDEF to process flight data. It can be used for test data as well, but the first option of specifying the few subsets needed for testing eliminates the need to get the EHS partial database files. The fields needed for this option are described below:

Directory
The Directory field is used to enter the directory where the EHS partial database download files reside.

Prefix
Filenames for EHS Partial Database files with telemetry information contain a prefix number. This number should be entered in the Prefix field.

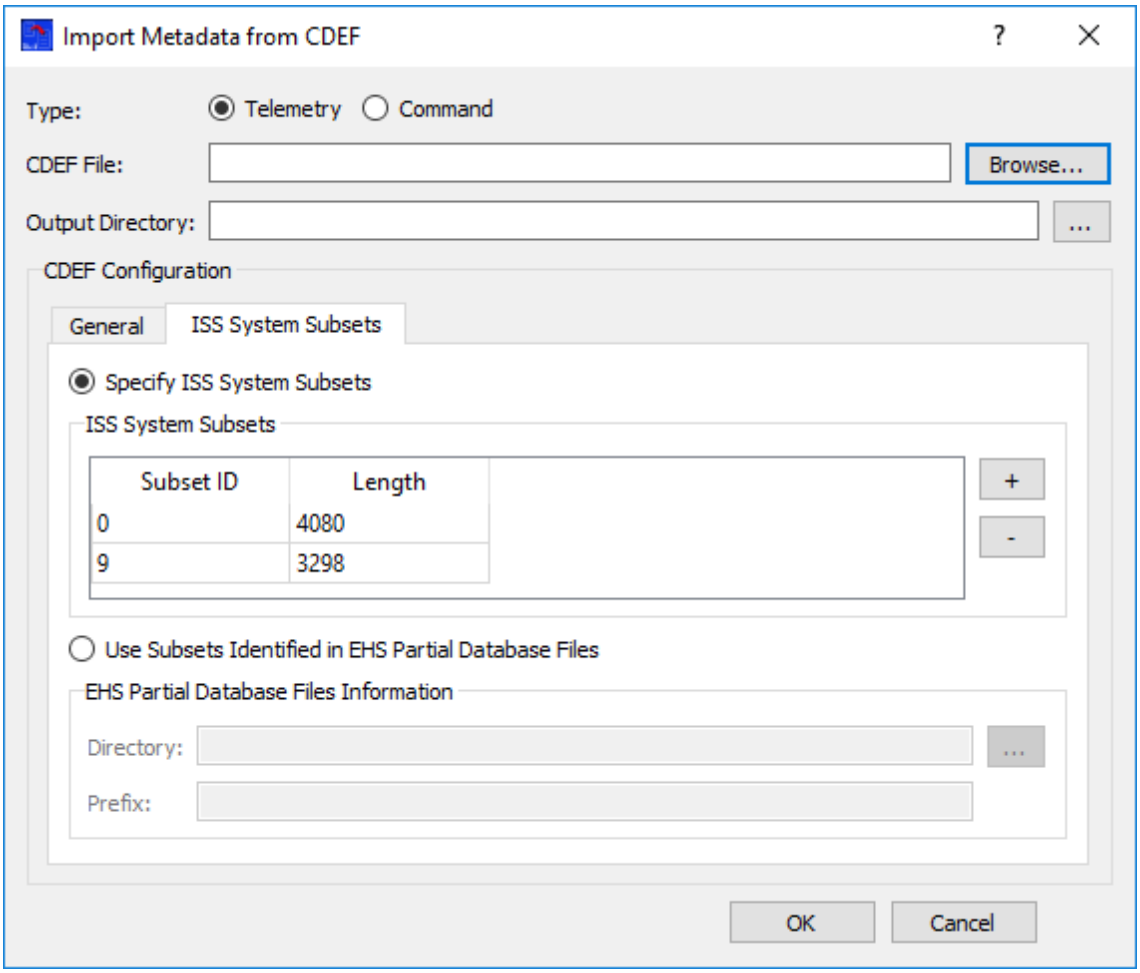


**Figure 66 Import Metadata from CDEF (ISS System Subsets Tab)**

## 6.30 Command Data Exchange Format Information

The capabilities provided by TReK are not a 100% match for the information that can be exchanged using CDEF. In some cases, CDEF has information that does not equate to something within TReK. In other cases, TReK provides additional capabilities that cannot be represented in CDEF. When information cannot be exported from TReK or imported into TReK, warning and error messages are generated as appropriate.

Warning messages are generated when the import or export code determines the item(s) identified in the message has the potential to cause information to be lost or modified.

Error messages are generated when the import or export code cannot successfully perform an action. This can be due to differences in capabilities supported by TReK and CDEF. The error wording should provide information on the item(s) causing the issues. In some cases, errors do not prevent files from being generated.

### 6.30.1 Export Differences

The following provides a high-level description of some capabilities in TReK that do not have a corresponding item in CDEF.

- TReK provides additional data types and lengths that are not supported by CDEF (e.g., 64-bit integers).
- Only ISS CCSDS and PDSS Payload packet types are supported by CDEF.
- CDEF limits the number of points allowed for spline calibration and the order of polynomial calibration.
- TReK allows names longer than the CDEF 13-character requirement. See the auto naming feature when exporting data to help meet this requirement.
- Other items such as descriptions have length limits. TReK will truncate descriptions as needed to meet this requirement. A warning message will be generated.

### 6.30.2 Import Differences

The following provides a high-level description of some items in CDEF that do not have a corresponding capability in TReK.

- Some data types cannot be represented in TReK. None of the data types are expected to be used by ISS payload developers as many are left over from support of the Spacelab and Space Shuttle programs. TReK will store these data types internally as either unsigned integers or undefined bytes but will restore the data type to the original upon export.

## 6.31 User Defined C Structure

If a C structure is used to define telemetry or command data in payload code, the C structure can be imported into TReK to define the metadata. A C structure is read into a Parameter Collection which can then be inserted into any Packet template to complete the definition. You can edit the Parameter Collection to add additional information as needed prior to inserting it into the Packet template.

***Allowed Types***

The allowed types of data within the structure are more restrictive than the types allowed by the TReK. Additional types can be supported upon request. Currently all data is assumed to be byte aligned (packed). The allowed data types and their corresponding data types and lengths in TReK are shown in the table below. Arrays are supported for all data types. If the translation to the TReK data type differs for an array, the array type is listed separately.

| C Type | TReK Data Type | Length (bits) |
|---|---|---|
| char | DT_TWOS_COMPLEMENT | 8 |
| char [n] | DT_NULL_TERMINATED_STRING | 8*n |
| double | DT_IEEE_FLOATING_POINT | 64 |

| | | |
|---|---|---|
| float | DT_IEEE_FLOATING_POINT | 32 |
| int | DT_TWOS_COMPLEMENT | 32 |
| int8_t | DT_TWOS_COMPLEMENT | 8 |
| int16_t | DT_TWOS_COMPLEMENT | 16 |
| int32_t | DT_TWOS_COMPLEMENT | 32 |
| int64_t | DT_TWOS_COMPLEMENT | 64 |
| long | DT_TWOS_COMPLEMENT | 32 |
| long long | DT_TWOS_COMPLEMENT | 64 |
| short | DT_TWOS_COMPLEMENT | 16 |
| uint8_t | DT_UNSIGNED_INTEGER | 8 |
| uint16_t | DT_UNSIGNED_INTEGER | 16 |
| uint32_t | DT_UNSIGNED_INTEGER | 32 |
| uint64_t | DT_UNSIGNED_INTEGER | 64 |
| unsigned char | DT_UNSIGNED_INTEGER | 8 |
| unsigned char [n] | DT_UNDEFINED_BYTES | 8*n |
| unsinged int | DT_UNSIGNED_INTEGER | 32 |
| unsigned long | DT_UNSIGNED_INTEGER | 32 |
| unsigned long long | DT_UNSIGNED_INTEGER | 64 |
| unsigned short | DT_UNSIGNED_INTEGER | 16 |

***Byte Order***

When importing a C structure into TReK you must specify how byte order of the data types within the structure should be interpreted. The byte order is for the system either sending the data to TReK or processing the data from TReK. The allowed byte order is big endian or little endian.

***How a C Structure is Interpreted***

The easiest way to show how a C structure is interpreted is with a simple example. Consider the structure below:

```
struct first
{
      int8_t v1;
      uint16_t v2;
      float v3;
};
```

The structure will be written to a Parameter Collection XML file based on the name of the structure. In this case the file will be “first.xml”. The Parameter Collection will consist of three parameters. Each parameter name will be the variable name (v1, v2, v3). A simple table shows some of the important attributes that are set for each parameter:

| **Name** | **Data Type** | **Length** | **Start Bit** |
|---|---|---|---|
| v1 | DT_TWOS_COMPLEMENT | 8 | 0 |
| v2 | DT_UNSIGNED_INTEGER | 16 | 8 |
| v3 | DT_IEEE_FLOATING_POINT | 32 | 24 |

***Enumerations***

You can use enumerated types in your structures. All enumerations are interpreted as 32-bit unsigned integers. Enumerations can be defined with or without values. The enumerations shown below are all valid:

```
enum my_enum{ one = 1, two, three=777 };

enum my_enum2 { four=4, five };

enum my_enum3
{
      six,
      seven
};

typedef enum
{
      eight,
      nine
} my_enum4;

typedef enum{ten,eleven,twelve}my_enum5;
```

***Nesting of Structures***

It is possible to nest structures. Nesting of structures produces names that include the nesting variables. Again a simple example is probably the best way to understand:

```
struct position
{
      double x;
      double y;
      double z;
};

struct locations
{
      struct position start;
      struct position end;
};
```

Two separate files will be produced in this example (position.xml and locations.xml). It is likely that the only XML file you really need is the top-level structure (locations.xml). TReK produces files for all structures found within a file.

| Name | Data Type | Length | Start Bit |
|---|---|---|---|
| start.x | DT_IEEE_FLOATING_POINT | 64 | 0 |
| start.y | DT_IEEE_FLOATING_POINT | 64 | 64 |
| start.z | DT_IEEE_FLOATING_POINT | 64 | 128 |
| end.x | DT_IEEE_FLOATING_POINT | 64 | 192 |
| end.y | DT_IEEE_FLOATING_POINT | 64 | 256 |
| end.z | DT_IEEE_FLOATING_POINT | 64 | 320 |

The resulting Parameter Collection no longer has the nesting of data. Each parameter will be placed at the same level and contain a concatenation of names based on the variables from each structure.

***Special Tags***

Special tags can be used in structure comments to override information or provide additional input for the resulting Parameter Collection. Each tag takes the form of:

```
TAG[data]
      TAG is the special tag name
      data is the value for the tag
```

The allowed tags are shown in the table below. Additional tags may be provided in a future release.

| Tag Name | Description |
|---|---|
| NAME | Overrides the variable name and uses the specified value as the name of the parameter |
| ALIAS | Sets the alias of the parameter to the value specified |
| DESC | Sets the short description of the parameter to the value specified. |
| FIXED | Sets the value for the parameter as specified and sets the modifiable flag to false. |
| MOD | Sets the value for the parameter as specified and sets the modifiable flag to true. |
| LOW | Sets a low range value as specified. |
| HIGH | Sets a high range value as specified. |

A short example of a structure will show how to use the special tags. The tags must appear on the same line as the variable.

```
struct start_health_and_status
{
      uint16_t cmd_id;     // NAME[CommandId] FIXED[2]
      uint16_t rate;       // DESC[Send data every n seconds] LOW[1] MOD[1]
      char ip_addr[16];    // NAME[DestIpAddress] MOD[192.168.1.6]
      uint16_t dest_port; // MOD[6200]
      uint16_t id;         // ALIAS[DownlinkId] MOD[3]
};
```

***Other Rules to Know***

There are some other things to consider when using this feature.

- The translator from the C structure to Parameter Collection is not a full C parser. It only knows a limited amount of information. It does not have the capability of looking in 'include' files so all the information about enumerations and structures must be defined in the same file.

- The translator currently does not support any of the standard C structures such as struct tm. These may be added in future releases as time permits or upon user request.

## 6.32 Application Messages

Various types of application messages are generated including information, progress, warning, error, and debug messages. Application messages are stored in memory and written to a temporary log file. The temporary log file is created on application initialization and exists if the application is running. It is deleted when you exit the application. The log file is in the temporary directory provided by the operating system. Only a subset of messages is stored in memory while all messages are written to the temporary log file. The maximum number of application messages stored in memory is controlled by the message storage setting in the Configure Messages dialog. Once the maximum is reached, older messages are deleted to make room for new messages. Setting the maximum value to a large number can impact application performance since it will increase the amount of memory used by the application. Setting this number too low can cause you to miss important messages. The application default was selected to protect against both scenarios. Messages stored in memory are displayed in the Main Window Message Area and the Messages dialog. The Messages dialog is shown in Figure 67. The Main Window message area only displays Info, Warning, and Error messages. The Messages dialog displays messages based on the display preferences defined in the Configure Messages dialog. By default, the Messages dialog will display information, progress, warning, and error messages. Columns in the Messages dialog can be sorted by clicking on the column header. The Messages dialog is available from the Options menu.

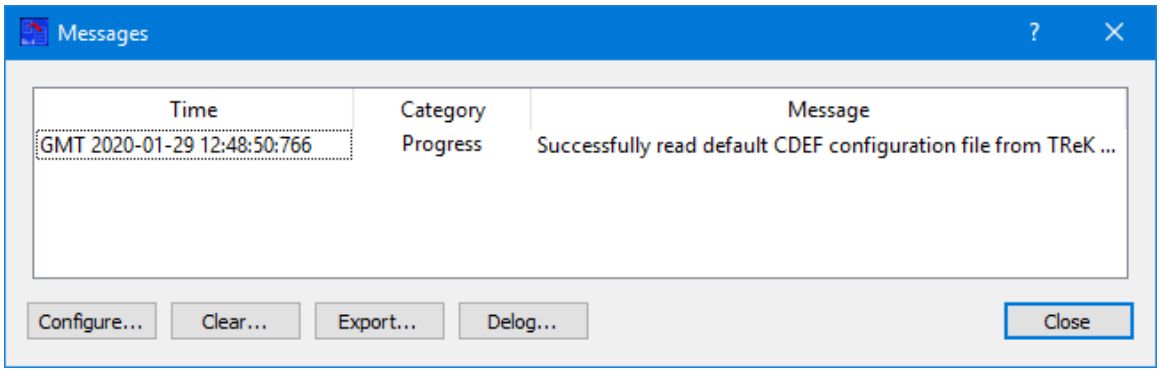


**Figure 67 Messages Dialog**

Configure

The Configure button provides access to the Configure Messages dialog shown in Figure 68. This dialog provides access to preferences associated with messages. Display preferences can be set to filter the types of messages (category) displayed in the

Messages dialog. Export Preferences control how the time tag is added to the filename that is created when messages are exported. See the Export section for details.

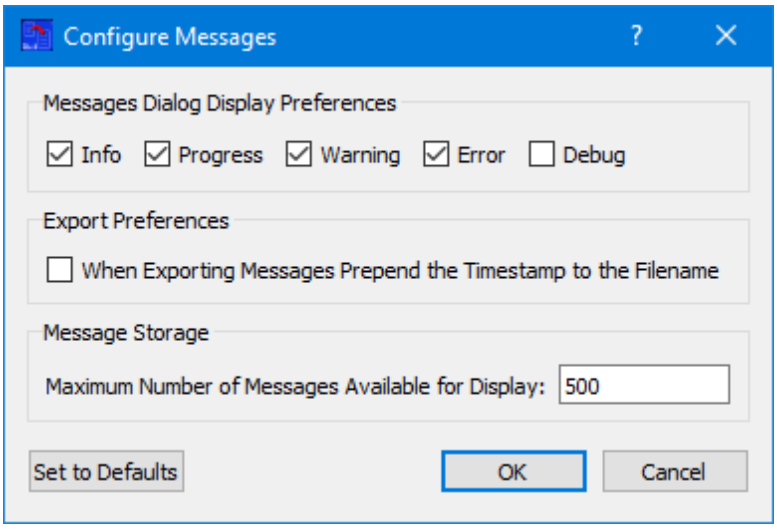


**Figure 68 Configure Messages Dialog**

Clear

The Clear button provides access to the Clear Messages dialog shown in Figure 69. This dialog provides two ways to clear application messages stored in memory. You can clear all the messages or clear selected messages. Once you clear messages, the messages are permanently deleted in all views (Main Window Message Area and the Messages dialog).

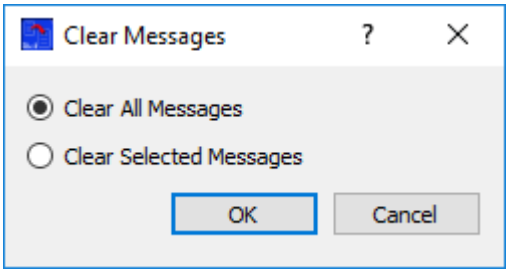


**Figure 69 Clear Messages Dialog**

Export

The Export button provides the capability to save all the application messages currently in memory to a file. When you push the Export button you will be prompted for a directory and filename. Export will save all messages in memory, not just the messages currently displayed in the Messages dialog (i.e., the Display Preferences are not applied). The name you provide for the file will be modified with a time tag that is added to the filename. The time tag indicates the time the file was closed. The default is to append the time tag to the filename. For example:

Filename Input: messages.txt
Filename Output: messages_2017-05-07_13~03~28.txt

If you would like to prepend the time tag to the filename you can set this preference in the Configure Messages dialog. This would result in the following:

Filename Input: messages.txt
Filename Output: 2017-05-07_13~03~28_messages.txt

Delog
The Delog button provides the capability to save all application messages generated since the application was started. Delog will retrieve the messages from the temporary log file. When you push the Delog button you will be prompted for a directory and filename. A timetag is not applied to the filename.

Filename Input: messages.txt
Filename Output: messages.txt

### 6.33 Application Configuration File

The Metadata application saves the information entered for each work area.

### 6.34 Application Settings

The Metadata application saves some settings as application settings each time you exit the application. The next time you run the application, the application will initialize with the previous application settings. The following application settings are saved:

- Application Window Size
- Configure Messages Selections

### 6.35 Application Command Line Arguments

The Metadata application accepts the following command line arguments:

- trek_metadata.exe <filename>

*filename* to open a configuration file (full path to file)

A value must be wrapped in double quotes if it contains spaces.

Examples:

- trek_metadata.exe “D:/metadata config.xml”

- trek_metadata.exe D:/metadata_config.xml

# 7 FAQ and Troubleshooting

This section addresses Frequently Asked Questions and provides tips for troubleshooting common gotchas.

No FAQs Yet.