ATML Workbench

Table of contents

Introduction

Thank you for your interest in the ATML Workbench.

This help document will provide you with instructions for how to use each of the tools. General information about the ATML Workbench and each of the tools is available in the rest of this section. Step by step instructions for how to use the tools are in the Typical Use section. Specific information about the Libraries, Translator, Reader, and Allocator are available in their respective sections of the help.

If you run into something that isn't covered in this document, please contact us.

Once again, welcome.

Welcome

Description

The ATML Workbench is an Integrated Development Environment (IDE) used in the creation and maintenance of ATML Documents. The application currently includes three primary tools: the Translator, the Reader and the Allocator. In addition, a custom Signal Model Library is available for license from UTRS. Automatic Test Markup Language (ATML) is an IEEE family of standards that creates a common format for the exchange of Automatic Test Equipment (ATE) and test information. ATML is based on eXtensible Markup Language (XML), which uses text files that are both human- and machine-readable to exchange structured data between applications.

ATML Workbench

The ATML Workbench is the primary container for each of the tools in the tool set. The ATML Workbench provides the user interface for each of the tools and associated libraries, as well as common services for the tools (e.g., messaging, file input/output). The ATML Workbench provides windows for each of the tools and selected output files. Each of the windows can be docked at a location of the user's choice.

Translator Tool

The Translator Tool is used to translate translates TPS source code from ATLAS into ATML. The current version of the Translator is targeted to CASS ATLAS TPSs. The Translator uses a two-step process, first creating ATLAS Intermediate XML (AIXML) and then creating the 1671.1 Test Description. The Test Description generated by the Translator specifies signal instances found in the ATLAS source using models drawn from a Signal Model Library.

Reader Tool

The ATML Reader tool is used to convert configuration files to 1671.4 Test Configuration files. Currently the Reader tool can read the Navy's TPSI file and automatically translate it directly to the ATML 1671.4 standard. Other file types may be opened, and utilizing advanced copy/paste, the user can manually create the Test Configuration.

Allocator Tool

The Allocator tool reads in the 1671.1 Test Description and the 1671.4 Test Configuration documents and determines which test stations selected by the user from the database are capable of performing the tests required. If a selected test station is unable to generate the required signals, the Allocator will indicate which capabilities are lacking.

Signal Model Library

The Signal Model Library is designed to contain IEEE 1641 Signal Test Definition (STD) compliant Test

Signal Framework (TSF) models. Libraries of signal models are available from IEEE, UTRS, and other organizations. In addition, users may develop their own signal models using commercially available software.

Next: Getting Started.

What's new

Version: 1.5

Date: 07/31/2015

- Initial release via github.com.

ATML Workbench Background

ATML Workbench Background

Test Program Sets (TPSs) are used in combination with automatic test equipment (ATE) to confirm that electronic systems and components are ready for issue (RFI) or to detect and isolate faults in the items. Generally, TPSs are designed to test specific units under test (UUT) using specific ATE. Modifications or improvements to either the UUT or the ATE can result in the requirement to modify the TPS as well.

The ATML Workbench is a suite of software tools designed to support the re-hosting of ATLAS TPSs to ATML. For a description of how to use the ATML Workbench, please see Typical Use. For a description of each tool within the ATML Workbench please see the Reader, Translator, or Allocator sections of this help.

Getting Started

Getting Started

This section of the help will allow you to quickly start using the ATML Workbench.

First, please make sure you have the appropriate System Requirements.

System Requirements

ATML Workbench System Requirements

Microsoft Windows 7, 8
Microsoft .NET 4.0
Microsoft Access Engine
Microsoft Office (for Office document previews)
Adobe Acrobat Reader

Next: Learn the Components.

Learn the Components

ATML Workbench Integrated Development Environment (IDE)

The ATML Workbench is the primary container for each of the tools in the tool set. The ATML Workbench provides the user interface for each of the tools and associated libraries, as well as common services for the tools (e.g., messaging, file input/output). The ATML Workbench provides windows for each of the tools and selected output files. Each of the windows can be docked at a location of the user's choice.

Project Menu

The Project Menu allows users to create, open, close, and delete projects; open files; and import and export Test Program Archives (TPARs).

View Menu

The View Menu allows users to select which standard windows are visible and provides access to the ATML Workbench Property Options interface screen.

Library Menu

The ATML Workbench provides users with access to a number of libraries. These libraries are databases designed to hold general documents, instrument descriptions, signal models, test adapter descriptions, test station descriptions, and Unit Under Test (UUT) descriptions. Additional information about libraries can be found here.

Reports Menu

The Reports Menu provides users with access to statistics and procedure hierarchy reports generated by the Translator.

Help Menu

The Help Menu provides users with access to log files and information about the ATML Workbench and each of the tools.

Admin Menu

The Admin Menu provides a mechanism for users to update help messages.

Project Navigator

The Project Navigator provides users with access to files associated with a TPS. It is initially located to the left of the screen, but can be repositioned by the user or hidden using the thumbtack icon in the upper right corner.

Main Window

The Main Window provides tabbed access to each of the tools and libraries. Each tab can be repositioned as a separate window.

Output Window

The Output Window displays messages, warnings, and errors to the user. It is initially located in the lower left corner of the screen and is tabbed to segregate messages from each of the tools.

Translator

The Translator Tool is used to translate translates TPS source code from ATLAS into ATML. The Translator uses a two-step process, first creating ATLAS Intermediate XML (AIXML) and then creating the 1671.1 Test Description. Users can trace from the Test Description back to the relevant ATLAS source code using the Source Comparison Tool. The Test Description generated by the Translator specifies signal instances found in the ATLAS source using models drawn from a Signal Model Library. Users can change the assignment of signal properties using the Signal Mapper.

1671.1 Test Description Window

This window displays the 1671.1 Test Description generated by the Translator. It is initially located at the lower right corner of the screen and appears when the Translator Tool tab is selected in the Main Window.

Reader

The Reader tool is used to convert configuration files to 1671.4 Test Configuration files. Currently the Reader tool can read the Navy's TPSI file and automatically translate it directly to the ATML 1671.4 standard. Other file types may be opened, and utilizing advanced copy/paste, the user can manually create the Test Configuration.

1671.4 Test Configuration Window

This window displays the 1671.4 Test Configuration generated by the Reader. It is initially located at the lower right corner of the screen and appears when the Reader Tab is selected in the Main Window.

Allocator

The Allocator tool reads in the 1671.1 Test Description and the 1671.4 Test Configuration documents and determines which test stations selected by the user from the database are capable of performing the tests required. If a selected test station is unable to generate the required signals, the Allocator will indicate which capabilities are lacking.

Instruments Window

The Instruments Window appears below the Adapters Window when the Allocator Tool is selected in the Main Window. It displays the instruments available in the library for users to select as a part of the Signal Analysis.

Stations Window

The Stations Window appears to the right of the Main Window when the Allocator Tool is selected in the Main Window. It displays the test stations available in the library for users to select as a part of the Signal Analysis.

Required ATE Window

This tabbed window appears in the lower right corner of the screen when the Allocator Tool is selected in the Main Window. It provides tabs to identify the test adapters, instruments, and signals identified as required by the test description and test configuration files.

Next: Typical Use.

Typical Use

Typical Use of the ATML Workbench

This section describes the typical order of events for using the ATML Workbench to translate an ATLAS TPS to ATML and conduct a signal analysis. It assumes that the libraries are populated with descriptions of Automatic Test Equipment (ATE) and signals of interest to the user.

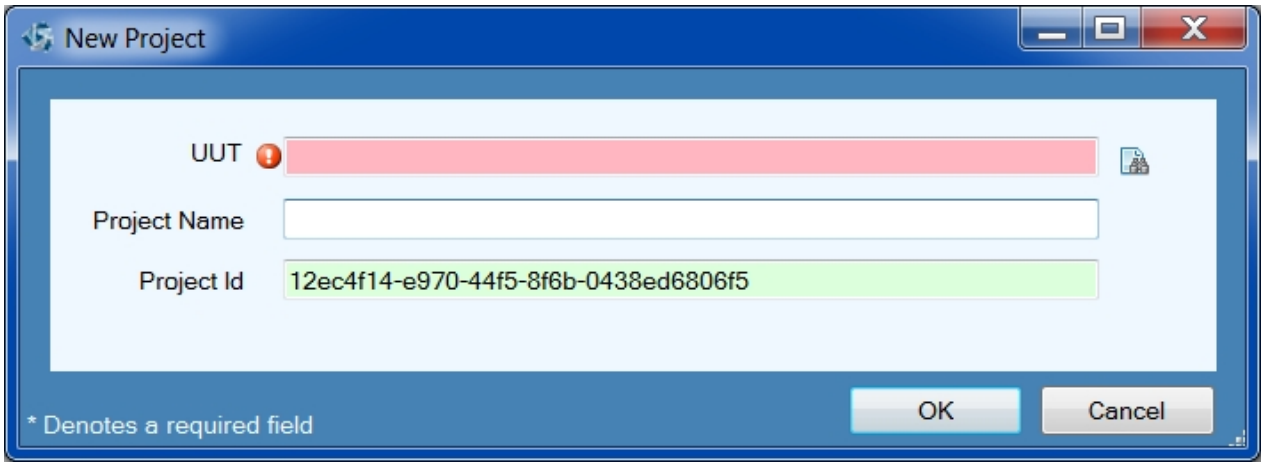
1. Create a New Project
2. Create a 1671.1 Test Description
3. Create a 1671.4 Test Configuration

4. Perform Analysis

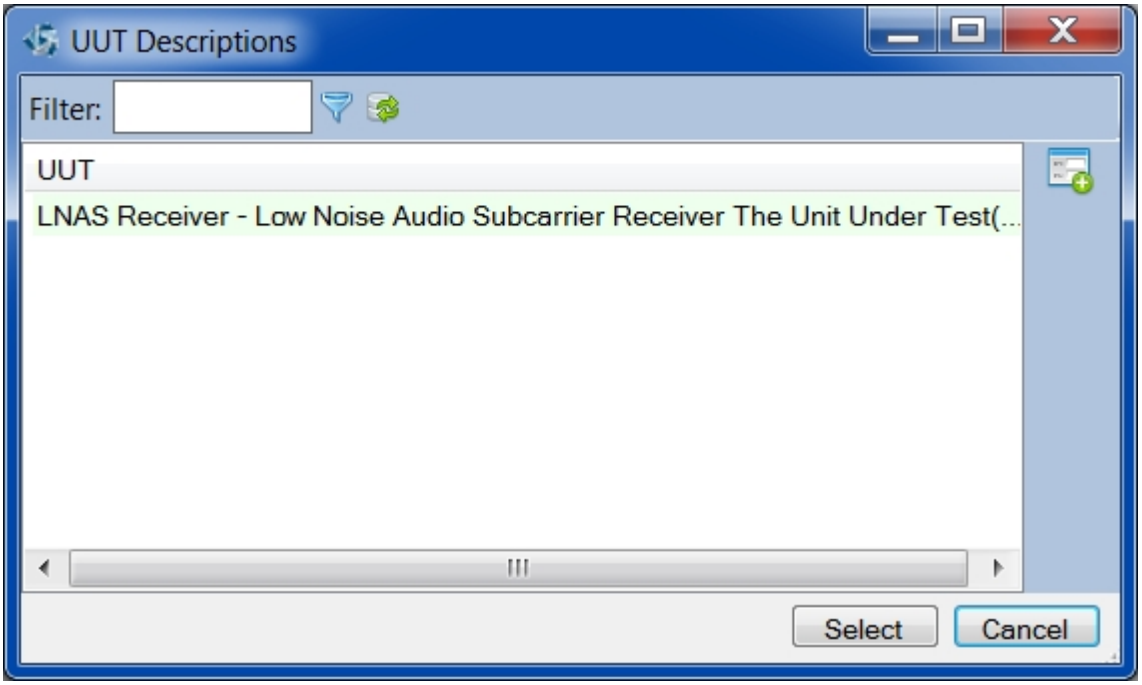
Create a New Project

Create a New Project

1. From the **Project** menu, select **New Project** or from the main toolbar press the button. The New Project form will appear:



2. Press the button to the right of the UUT edit control. The UUT descriptions window will appear. This window allows users to choose a UUT from the entries in the UUT Descriptions Library.



3. Select the UUT associated with the project. By typing a partial name in the filter box, the number of entries in the list will be limited to only those UUTs containing the filter text in its name.

   If the UUT required is not in the list you may create a new one by pressing the button.

4. Once you select the required UUT press the "**Select**" button. You will be returned to the New

Project form.

5. The Project Name field is automatically populated based on the UUT selected. You may edit this name if you choose. Typically you would use the UUT's name or the name of the original TPS that you are translating. Note that special characters associated with filenames (e.g., "/" or "*") are not allowed in the project name.

6. Each project requires a unique identifier. This identifier must be a UUID. The Project ID is automatically populated.

7. After each of the fields are filled in, press the "**OK**" button. Upon pressing the button the Project Navigator will be populated with the appropriate folders.

8. Next step: Create a 1671.1 Test Description.

Create a 1671.1 Test Description

Create a 1671.1 Test Description

The Translator Tool can be used to create a 1671.1 Test Description based on ATLAS source code by following these steps:

1. Add Source Documents

2. Run the AIXML Generator

3. Map Signals to TSF Libraries

4. Run the AIXML Translation

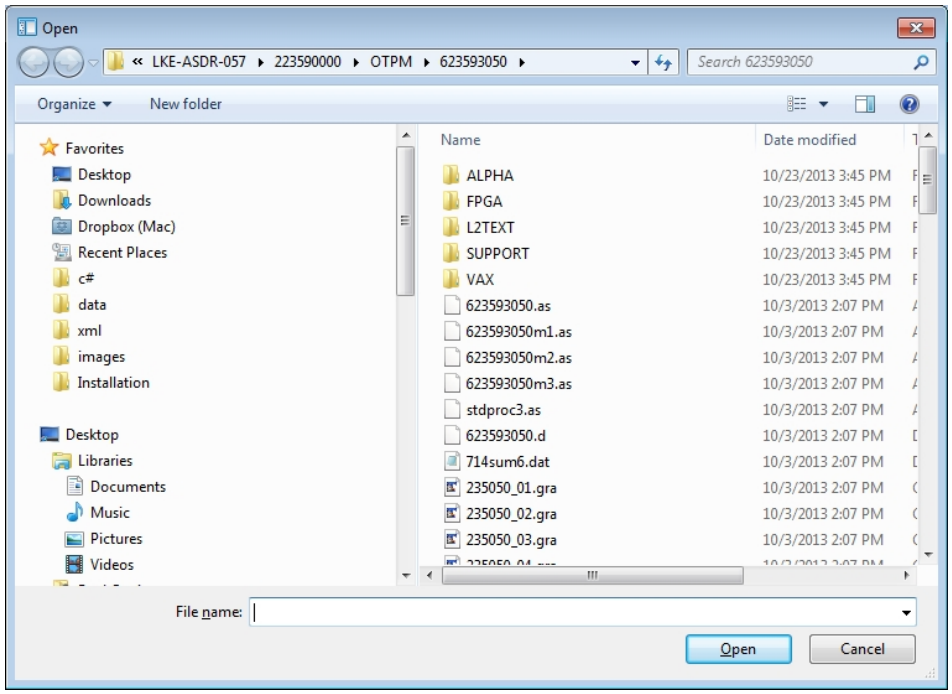
Add Source Documents

Add Source Documents

The ATLAS source code must be added to the "translator/source" folder in the Project Navigator so that the Translator can access them.

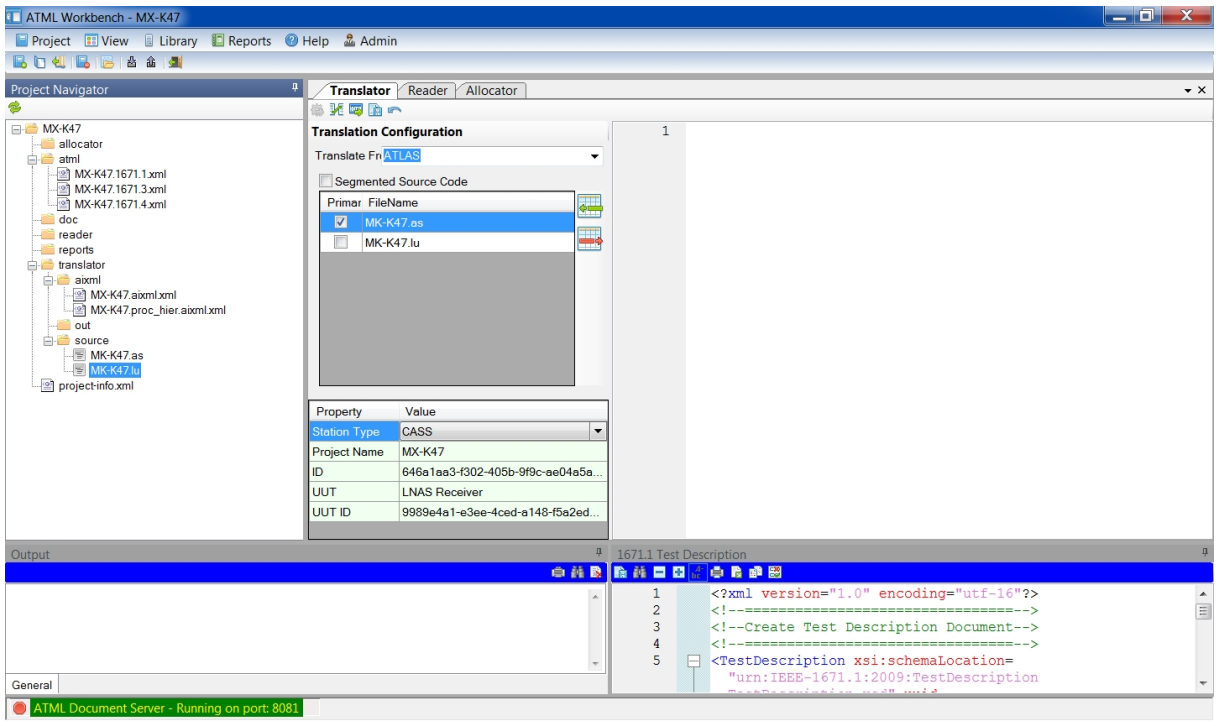
1. Right click on the "source" folder and select "Add Files" from the pop-up menu

2. A File Open dialog box will appear.



3. Navigate to the folder holding your CASS ATLAS source code and select each file. This should include supporting files such as lookup (LU) files. Multiple files may be selected using the Ctrl or Shift keys when clicking on file names. When the files are selected, click the Open button in the dialog box to add the files to the "translator/source" folder.

4. Files will be automatically populated to the Translation Configuration Interface. You may also add files either by selecting them in the source folder and dragging and dropping them into the gray box of the Translation Configuration Interface or by pressing the  button which will open a file selection form to allow a file to be selected from the network and added to the list. To remove a file from the list simply select the file to remove and press the  button



5. If your source file consists of "Segmented Source Code" check the appropriate box below the Translate From drop down.

6. Select the primary file by checking the box next to that file's name. When a filename is checked, it will automatically move to the top of the list.

7. Also when dealing with Segmented Source Code, the order that the files are processed is critical. The Translator will attempt to put the files in the correct sequence but you can change the order. When the Segmented Source Code box is checked, two additional buttons,  Move Down and  Move Up, will appear to allow you to reorder the files in the list.

8. Select the Station Type using the drop box list located towards the bottom of the Translation Configuration Interface.

9. When completed you must save the configuration by pressing the **Save** button on the Translator toolbar 

.

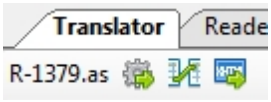
10. To undo any changes press the button.

Next: Run the AIXML Generator.

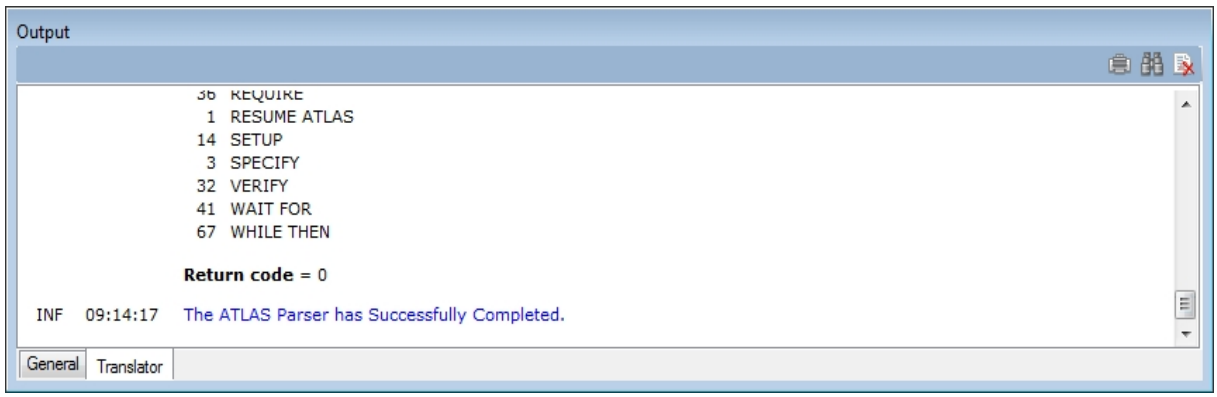
Run the AIXML Generator

Run the AIXML Generator

Once the source files have been added and the translation configuration completed (see Add Source Documents) you can now begin the translation.

1. Click on the Parse Source Document button from the Translator tool bar 

2. The Translator tab will appear in the Output Window at the lower left corner of the screen to provide ongoing updates as the parsing proceeds.



3. Once completed, the focus will shift to the AIXML file opened in its' own document tab. The file will

have the name of the project concatenated with .aixml.xml.

Next: Map Signals.

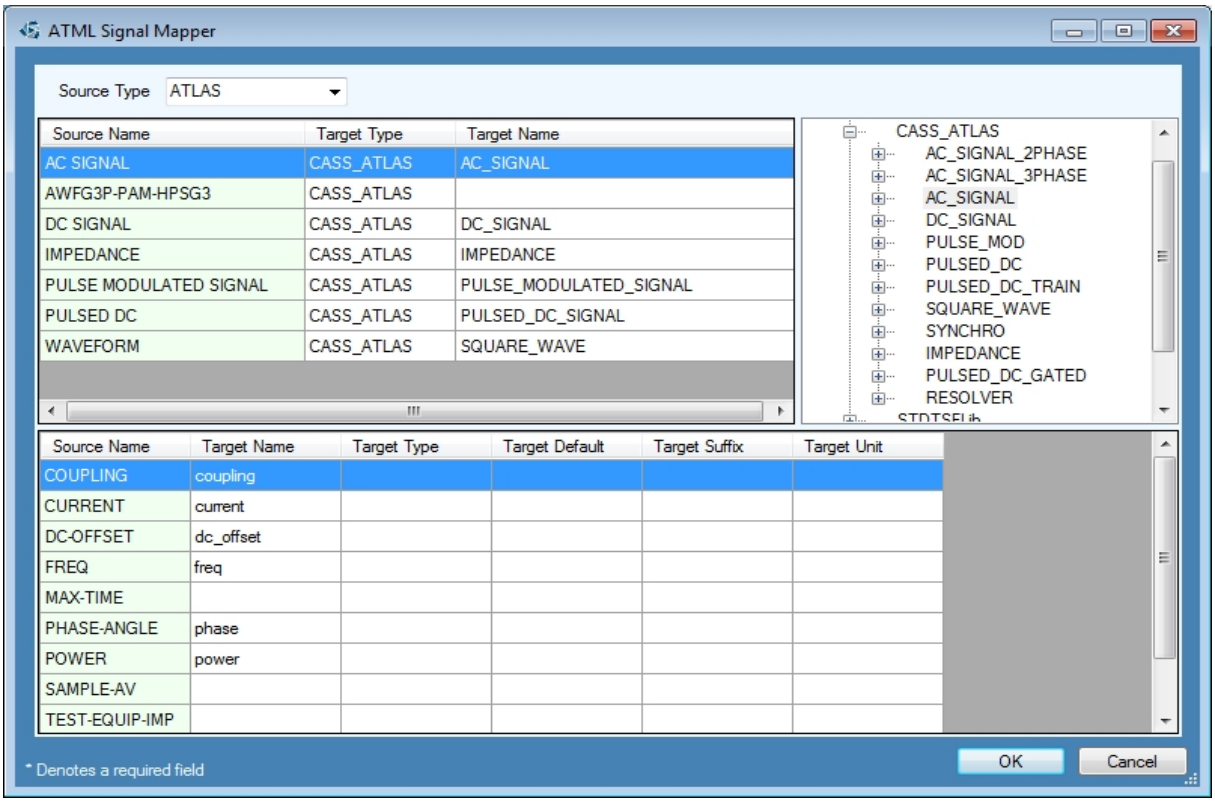
Map Signals to TSF Libraries

Map Signals

Once the source files have been translated to AIXML (see Run the AIXML Generator) you can run the ATML Signal Mapper. Ensure the focus is on the Translator Tab.

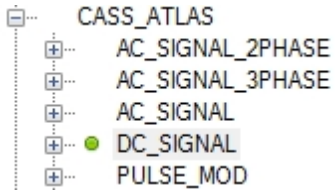
1. Click the Build Signal Map button  available on the Translator tool bar 

2. The ATML Signal Mapper screen will open.



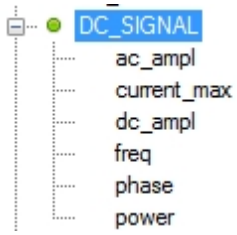
3. The Signal Mapper is divided into three sections. The top left section shows the source signal name and the selected target signal names. Directly under the source signal section is the source signal attribute sections. This section shows the attributes for the selected source signal as well as the selected target signal attributes. Directly to the right of the source signal section is the TSF Library tree which contains a list of all the available TSFs in the Signal Model Library.

4. As source signals are selected, the application will search the TSF tree for the best match based on the source signal attributes and mark that TSF with a green dot.



5. To map a TSF signal to a source signal simply select the TSF signal and drag it onto the source signal Target Name of the signal you want to map. The application will then attempt to map the correct signal attributes based on the attribute names.

6. Because names of attributes may be different between the TSF and the source signal attributes, you may have to directly map the attributes. To do this open the TSF signal name by clicking the + next to the selected signal name.



7. You can now highlight the TSF attribute name and drag and drop the attribute onto the source signal attribute Target Name of the source signal attribute you would like to map.

8. A 100% mapping is desired but not necessary, however the more data available the better the translation and analysis.

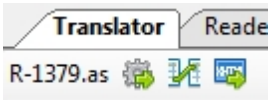
Next: Run the ATML Generator.

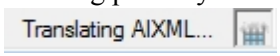
Run the ATML Generator

Run the ATML Generator

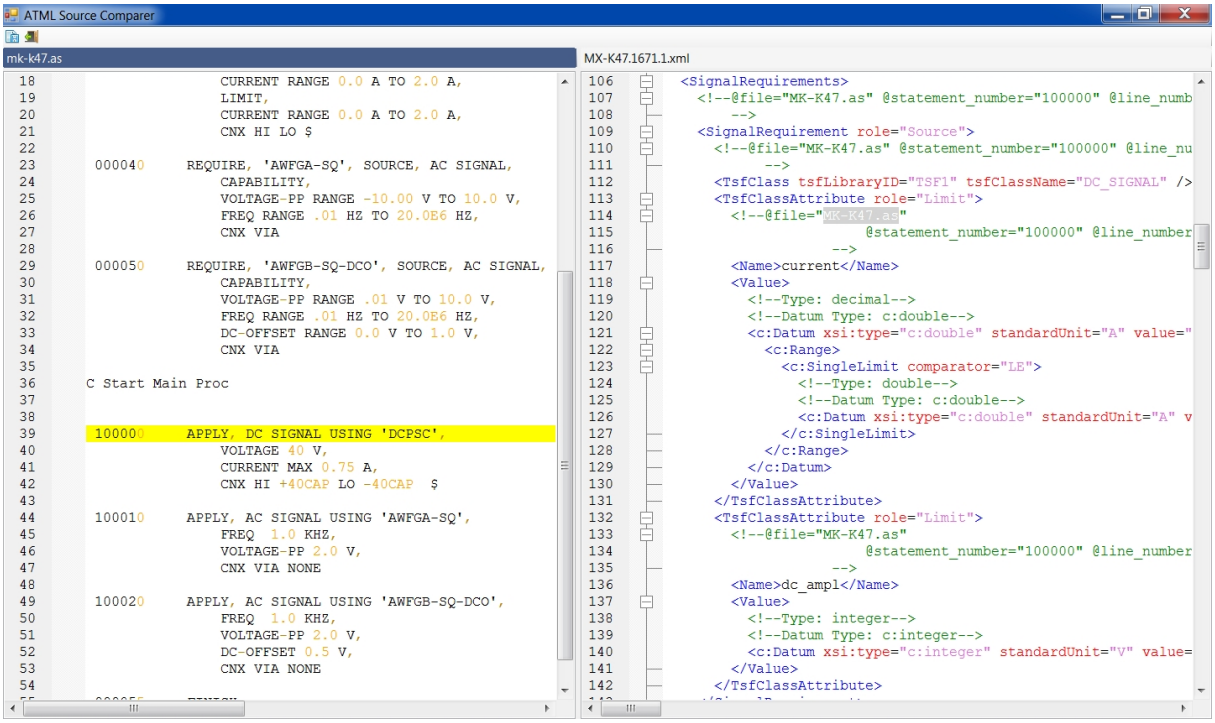
Once the source signals have been mapped (see Map Signals to TSF Libraries) you can now run the ATML Generator.

1. Click the AIXML to ATML Translation button  available on the Translator tool bar



2. The translation process will begin and informational updates will be provided in the Output Window. Because some translations may take a long time to complete, the translation will be performed in the background allowing you to proceed with other work within the ATML Workbench. It should be noted though that during translation some functionality will be disabled in order to permit the current translation to proceed. You will know a translation is taking place by an animation shown on the far right of the status bar at the bottom of the screen 

3. Once the translation is complete the animation will disappear and the ATML Source Comparison Tool screen will appear (see ATML Source Comparison Tool). This tool allows you to double click on a comment beginning with @file (in green) and be directed to the corresponding ATLAS source code for that portion of ATML code.



Next: Create a 1671.4 Test Configuration File.

Create a 1671.4 Test Configuration

Create a 1671.4 Test Configuration File

The Reader Tool can be used to create a 1671.4 Test Configuration file by following these steps:

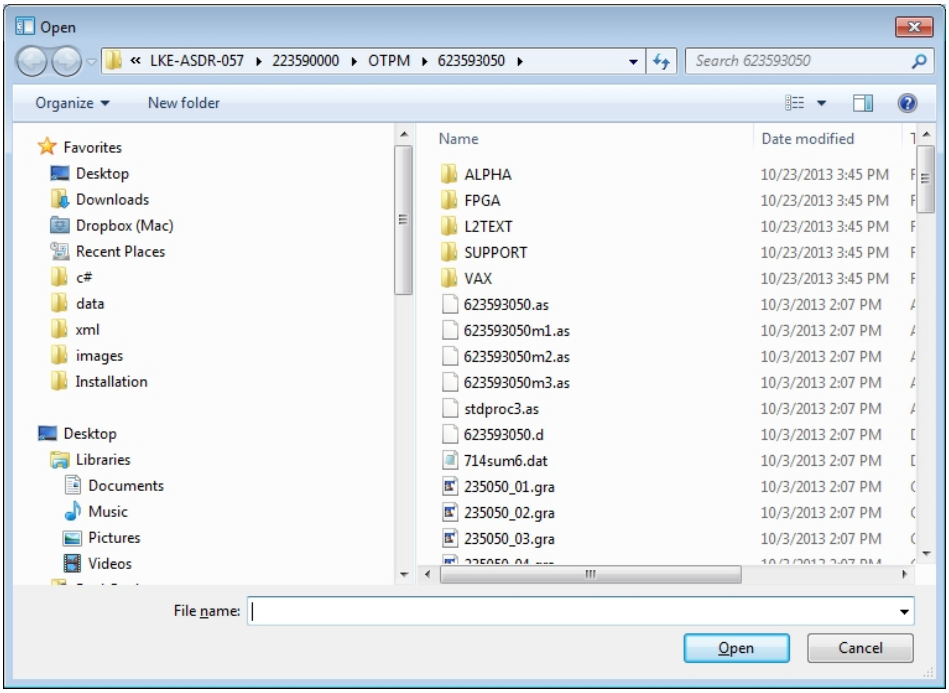
1. Add Legacy Configuration Files
2. Automatically Extract Configuration Data
3. Manually Enter Configuration Data

Add Source Configuration Files

Add Source Configuration Files

The source configuration files must be added to the "reader" folder in the Project Navigator so that the Reader can access them.

1. Right click on the "reader" folder and select "Add Files" from the pop-up menu
2. A File Open dialog box will appear.



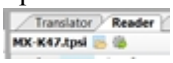
3. Navigate to the folder holding your source documents and select each file. Multiple files may be selected using the Ctrl or Shift keys when clicking on file names. When the files are selected, click the Open button in the dialog box to add the files to the reader folder.

4. The Reader can open files that are MS Word, MS Excel, Adobe Acrobat, text or rich text, and .tpsi. The .tpsi files can be used as the basis to automatically create a 1671.4 Test Configuration file. The other file types can be used to manually enter test configuration data using copy and paste.

Next: Automatically Extract Configuration Data.

Automatically Extract Configuration Data

Automatically Extract Configuration Data

The Reader Tool can automatically extract configuration data from .tpsi files. Simply open the file from the "reader" folder and click on the "Parse Input Document" button 

on the Reader toolbar. If any of the asset identifiers are not already in the database, the ATML Workbench will prompt you to automatically add or skip them using a dialog box. Assets added in this way will be "stubbed out" in the database, i.e., a document of the appropriate type and an associated UUID will be added to the database.

Once all available data has been extracted, the Reader will add the data to the proper field in the Test Configuration database and create the 1671.4 Test Configuration file.

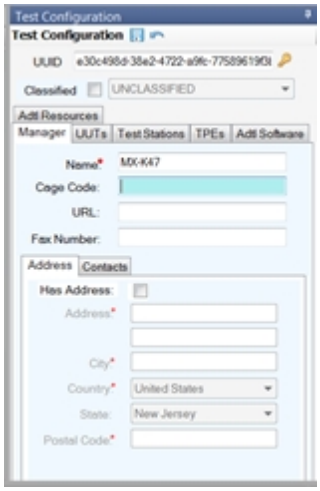
If you do not have a .tpsi file or if the file did not contain all desired data, you may add additional data (see Manually Enter Configuration Data.)

Manually Enter Configuration Data

Manually Enter Configuration Data

In order to manually enter configuration data, you may simply click in the appropriate field in the Test Configuration Window.

If the desired data is in a file that the Reader can open, you may use the advanced copy and paste function of the ATML Workbench to populate the fields. Open the file with the data using the Reader. Click in the field of interest so that it turns blue as seen below in the CAGE Code field.



Then hold down the Ctrl key and select the text from the file. The text will be automatically pasted into the correct field in the Test Configuration Window.

Be sure to click the Save button on the Test Configuration toolbar as you work.

Once a Test Description and Test Configuration are completed, the Allocator may be used to evaluate the test requirements against ATE capabilities (see: Perform Analysis).

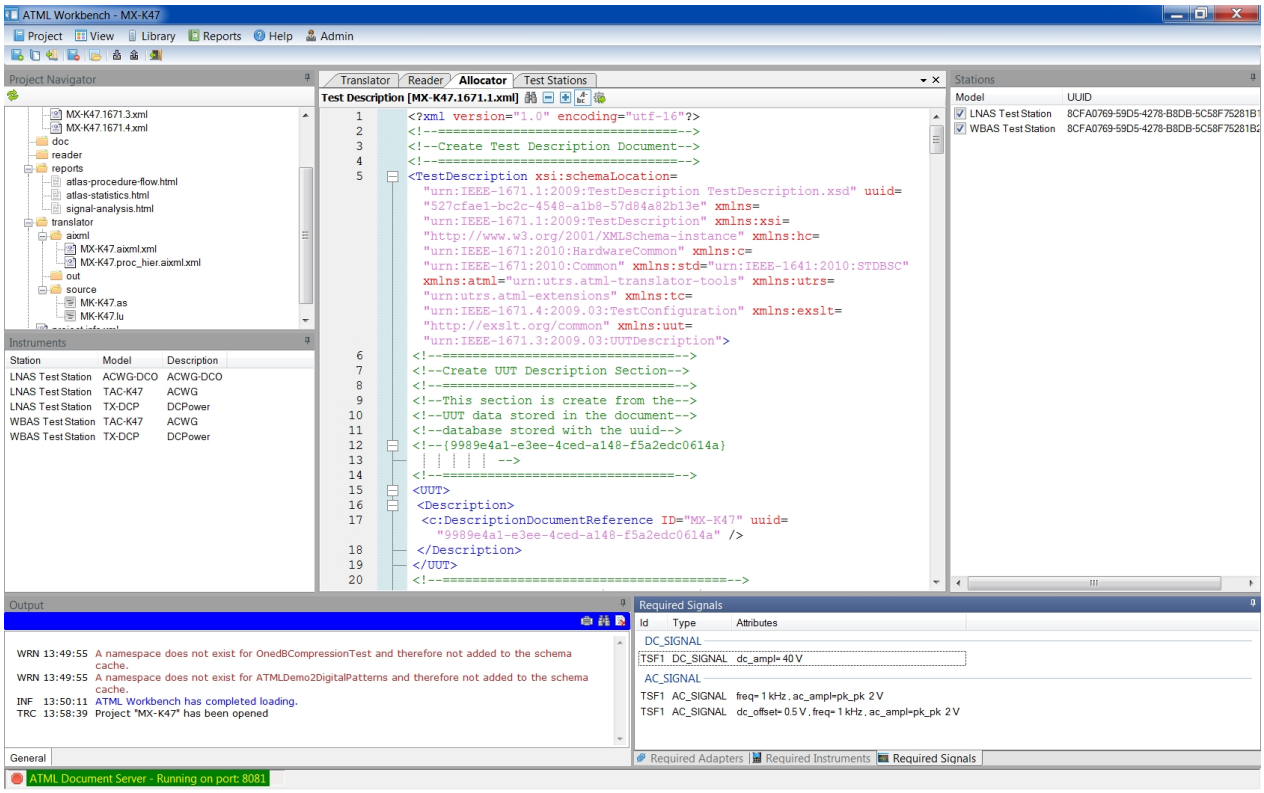
Perform Analysis

Perform Analysis

To use the Allocator, your project must have a completed 1671.1 Test Description and a completed 1671.4 Test Configuration file in the "atml" folder.

In addition, you must have completed descriptions of the test stations, instruments, and test adapters that you wish to evaluate.

When you open the Allocator tab, the test description will be opened in the main window. The required adapters, required instruments and required signals based on the data in the test configuration file will be presented in the tabbed window at the lower right corner of the screen.



The Stations window at the right of the screen presents the test stations that are in the ATML Workbench's database. You may select one or more of the stations for the analysis. The Instruments Window at the left of the screen between the Project Navigator and the Output Window will be populated with instruments based on the test station(s) selected. If you click on the signals in the Required Signals tab, the Allocator will highlight the instruments and test stations capable of producing the required signal.

To conduct a signal analysis, select stations and click the Perform Analysis button on the Allocator tool bar 

.

The signal analysis will compare the signals required to the capabilities defined in the Test Station description documents.

The signal analysis will generate a report in a new tab that indicates whether the selected station is fully capable, partially capable, or not capable of generating the signals required by the test description.

Getting Help

Getting Help

We hope that you will find using the ATML Workbench an intuitive and bug-free experience. However, should you need additional help or should you encounter a bug, please contact us at ATMLWorkbench@utrs.com.

FAQ

ATML Workbench Frequently Asked Questions

What is a project?

A project is a collection of documents all relating to a Test Program Set (TPS). These documents will include all existing ATLAS source files, Lookup (LU) files, TPSI files and any other documentation relevant to the test or UUT being tested. The files must be added to the appropriate folders so they can be utilized by each of the tools within the ATML Workbench. Details on the folder structure can be found in the "Project Navigator" section of this document.

How do I create a new project?

See the Create a New Project section of this Help for detailed instructions.

How do I import an existing project?

Existing projects may be imported using the Import TPAR action item from the Project Menu.

How do I export a project?

You may export a project using the Export TPAR action from the Project Menu.

Where do I put my source files?

If your source files are ATLAS code, they should be placed in the "translator/source" directory (see Add Source Documents).

If your source files are .tpsi or other files containing test configuration data, they should be placed in the "reader" folder (see Add Source Configuration Files).

Where can I find more libraries?

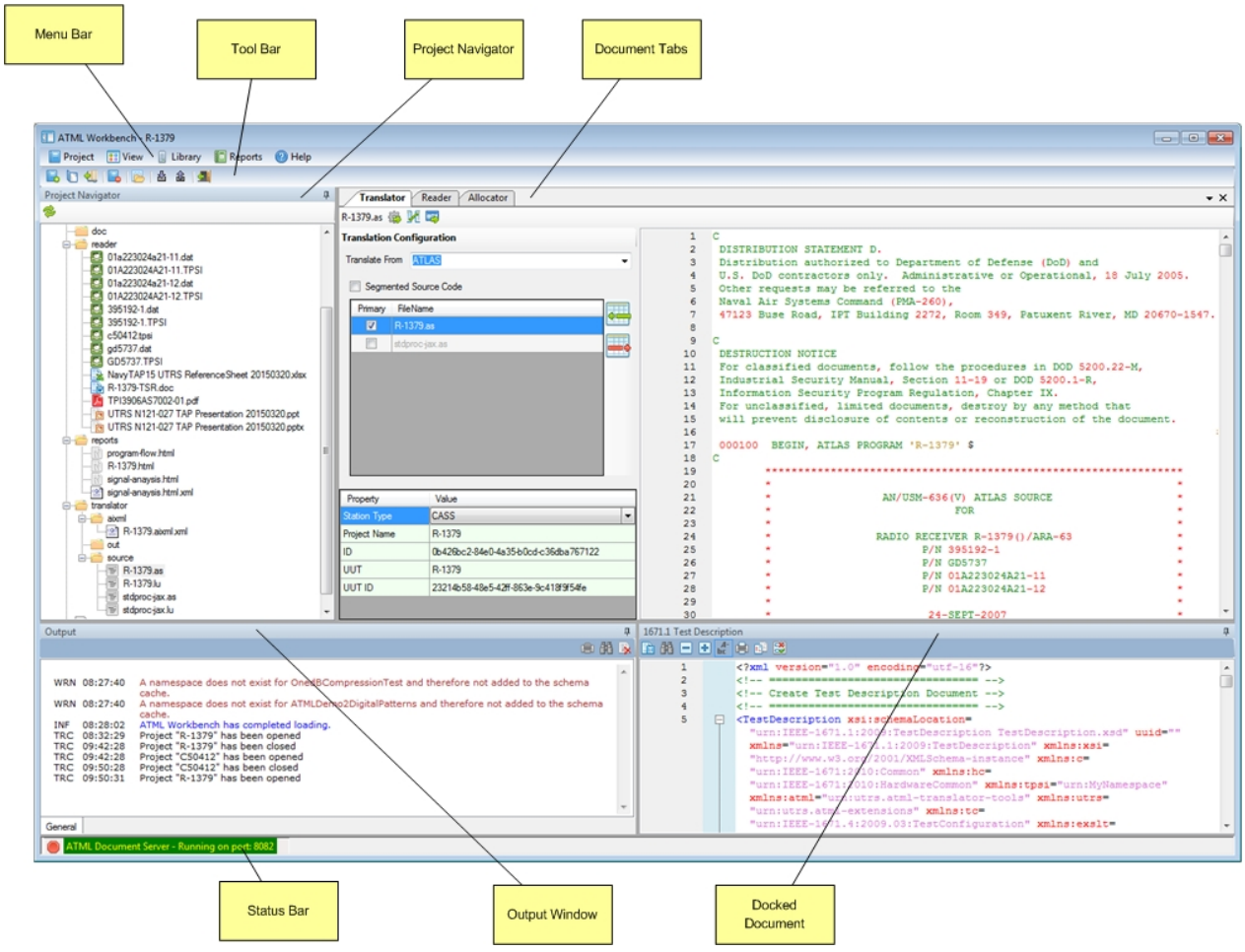
Where can I find more Libraries?

The ATML Workbench comes with signal model libraries created for IEEE ATML demonstration projects and with example documents for each of the ATML document types.

A signal model library specific to CASS TPSs is available for license from UTRS. In addition, UTRS would be happy to help you develop signal models specific to your application or to develop descriptions of your ATE. Please contact us at ATMLWorkbench@utrs.com.

ATML Integrated Development Environment (IDE)

Integrated Development Environment (IDE)



The IDE is comprised of the following components:

Menu Bar

Menu Bar

The Menu Bar is a collection of actionable items for working within the ATML Workbench. The menu bar consists of the following categories of action items:

Project

The Project menu is a collection of Project related action items.

New Project
This action item is used to create a new project.

Open Project
An action item used to open an existing project.

Close Project
An action item used to close the currently opened project.

Delete Project
An action item used to delete/remove the currently opened project.

Open File
An action item used to open a file that may not be directly related to a project.

Import TPAR
An action item used to import a Test Program Archive (TPAR).

Export TPAR
An action item used to export the currently opened project as a Test Program Archive (TPAR).

Exit
An action item used to close the ATML Workbench.

View

A collection of action items related to the visible windows of the ATML Workbench.

Edit Options

An action item used to open the ATML Workbench properties window.

Project Navigator

An action item used to hide or show the Project Navigator window.

Output Window

An action item used to hide or show the Output window.

Translator

An action item used to hide or show the Translator tool window.

Reader

An action item used to hide or show the Reader tool window.

Allocator

An action item used to hide or show the Allocator tool window.

Library

The Library menu option contains a collection if library related action items.

Documents

An action item used to show or hide the Document Library widow. The Document Library provided the means to maintain all the documents stored within the workbench. This includes all common ATML documents as well as all common documentation and other supporting application documents.

Instruments

An action item used to show or hide the Instrument Library. The Instrument Library is a view into the Document Library where only Instruments are shown. Performing maintenance on an instrument from this library will utilize the appropriate data collection forms rather than the document editor used when editing from the Document Library.

Signals

Signals is an action item used to show or hide the Signal Model Library. The Signal Model Library is a read only view of each of the TSF library documents used within the ATML Workbench. TSF library documents may be added as needed to work with different projects.

Test Adapters

An action item used to show or hide the Test Adapter Library. The Test Adapter Library is a view into the Document Library where only Test Adapters are shown. Performing maintenance on test adapters from this library will utilize the appropriate data collection forms rather than the document editor used when editing from the Document Library.

Test Stations

An action item used to show or hide the Test Station Library. The Test Station Library is a view into the Document Library where only Test Stations are shown. Performing maintenance on test stations from this library will utilize the appropriate data collection forms rather than the document editor used when editing from the Document Library.

UUTs

An action item used to show or hide the UUT Library. The UUT Library is a view into the Document Library where only UUTs are shown. Performing maintenance on UUTs from this library will utilize the appropriate data collection forms rather than the document editor used when editing from the Document Library.

Reports

The Reports menu contains a collection of report related action items.

Status Report

An action item used to run and display a status report pertaining to the currently opened project.

Help

The Help menu option is a collection of help related action items.

Log Files

This action item provides the means to open an application log file. Log files are created on a daily basis and therefore a prior day's log may be opened. Logging is only performed if the logging option is set to the affirmative in the application properties window (see View).

About

This action item opens a document outlining information pertaining to the ATML Workbench application, such as 3rd party licenses use.

View Help

This action item is used to open this help document.

Release Notes

Release Notes is an action item used to open a document highlighting the major changes for each released version of the ATML Workbench.

Admin

The Admin menu allows users to perform administrative tasks.

Help Messages

This action item allows users to import updated help messages or export messages currently in use.

Tool Bar

Tool Bar

The Tool Bar is a collection of shortcut buttons used to access the functionality of those action items listed under the Menu Bar. By hovering the mouse pointer over a button will display a tool tip explaining the operation of the shortcut button.

 **Add New Project**

 **Open Project**

 **Close Project**

 **Delete/Remove Project**

 **Open File**

 **Import Project Archive (TPAR)**

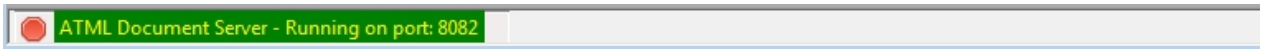
 **Export Project Archive (TPAR)**

 **Exit The Application**

Status Bar

Status Bar

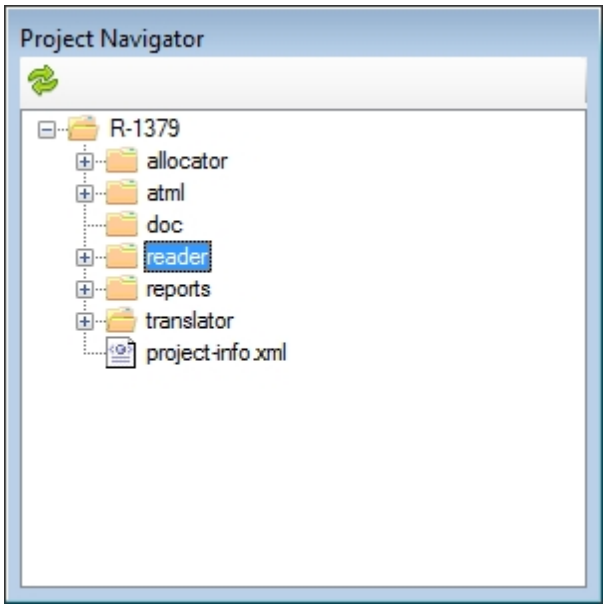
The Status Bar shows the current health of the ATML Workbench - particularly the HTTP document server. A button to the left of the status provides the ability to stop and start the document server. Currently this document server is not required but may be necessary for future use - particularly in a networked or peer to peer scenario of document management.



Project Navigator

Project Navigator

The Project Navigator is a tree list view of the files and folders associated with a project. The navigator consists of the root folder which has the name of the project itself, and 5 sub-folders, one for each tool, one for the ATML output documents, and one for general documentation.



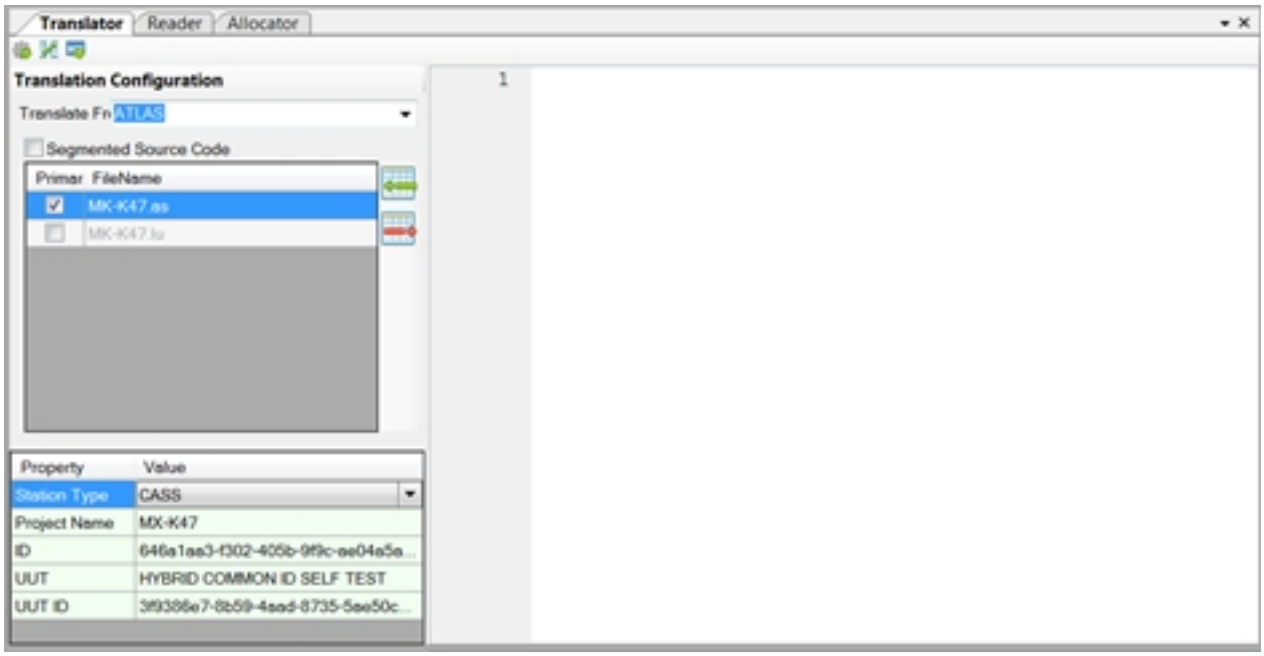
Folder Structure

allocator	Contains any documents related to the Allocator tool.
atml	Contains any project-related ATML documents such as the 1671.1 test description or the 1671.4 test configuration files.
doc	Contains any project related documents.
reader	Contains any documents related to the Reader tool such as the TPSI file. Documents added to this folder will be opened within the Reader.
reports	Contains reports and data files created by the ATML Workbench.
translator	Contains all Translator related documents such as TPS source code and related files. The translator folder has several sub folders.
aixml	Contains the ATLAS Intermediate XML file generated by the Translator tool.
out	Contains any work files generated by the Translator tool (note, the Translator will cleanup these files by default).
source	Contains any source code TPS files. This includes ATLAS source documents as well as lookup (LU) files.

Main Window

Main Window

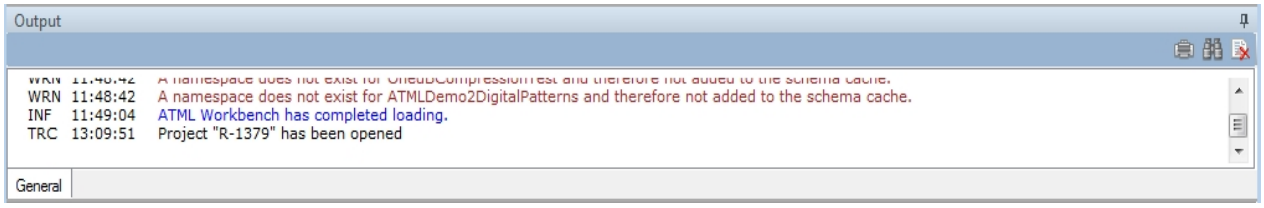
The Main Window of the ATML Workbench is the primary/default docking area for each of the tools as well as any opened documents. Any of the tools or documents may be detached from this area and either re-docked in another area or kept "floating" as a standalone window.



Output Window

Output Window

The Output Window displays all the messages presented from each of the tools as well as from the IDE itself. Messages from the tools will be displayed in an output window tab specific to the tool. IDE messages or common application messages will be displayed in the "General" tab. Text within the output window can be searched, printed or cleared using the tool buttons provided.



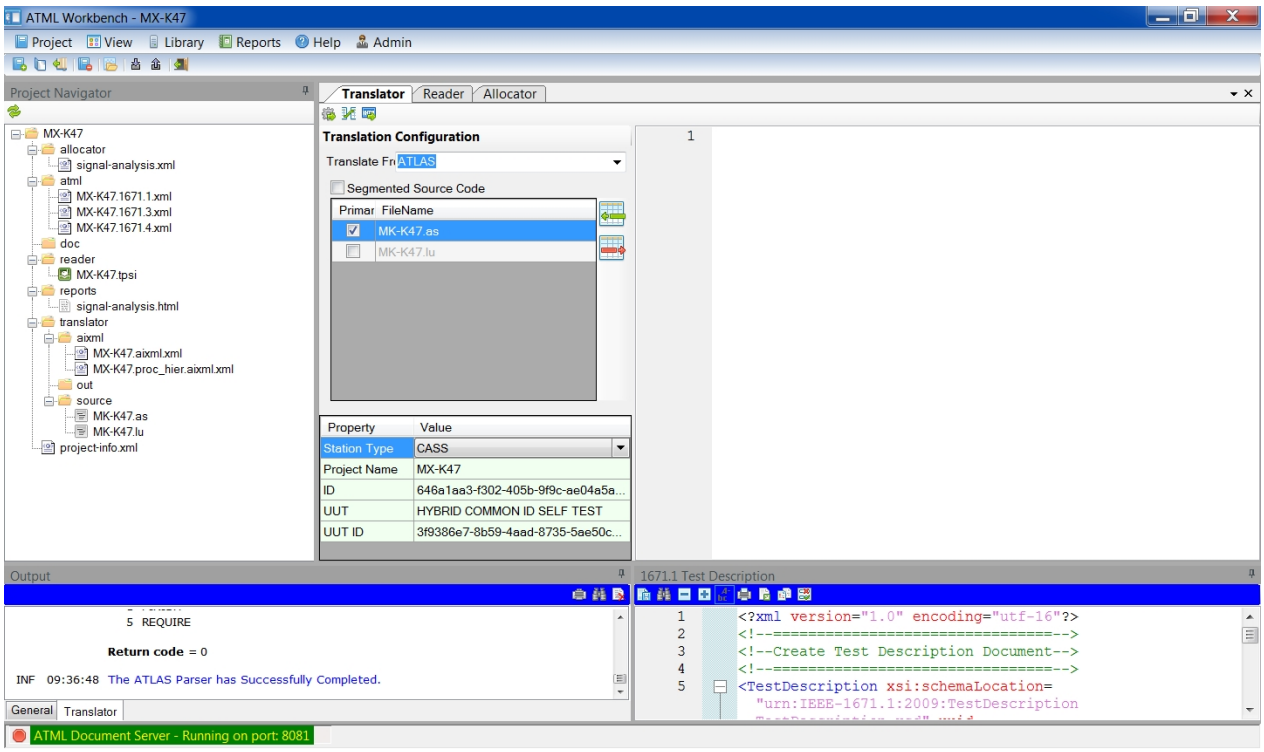
The severity of the message is presented using both color and acronym:

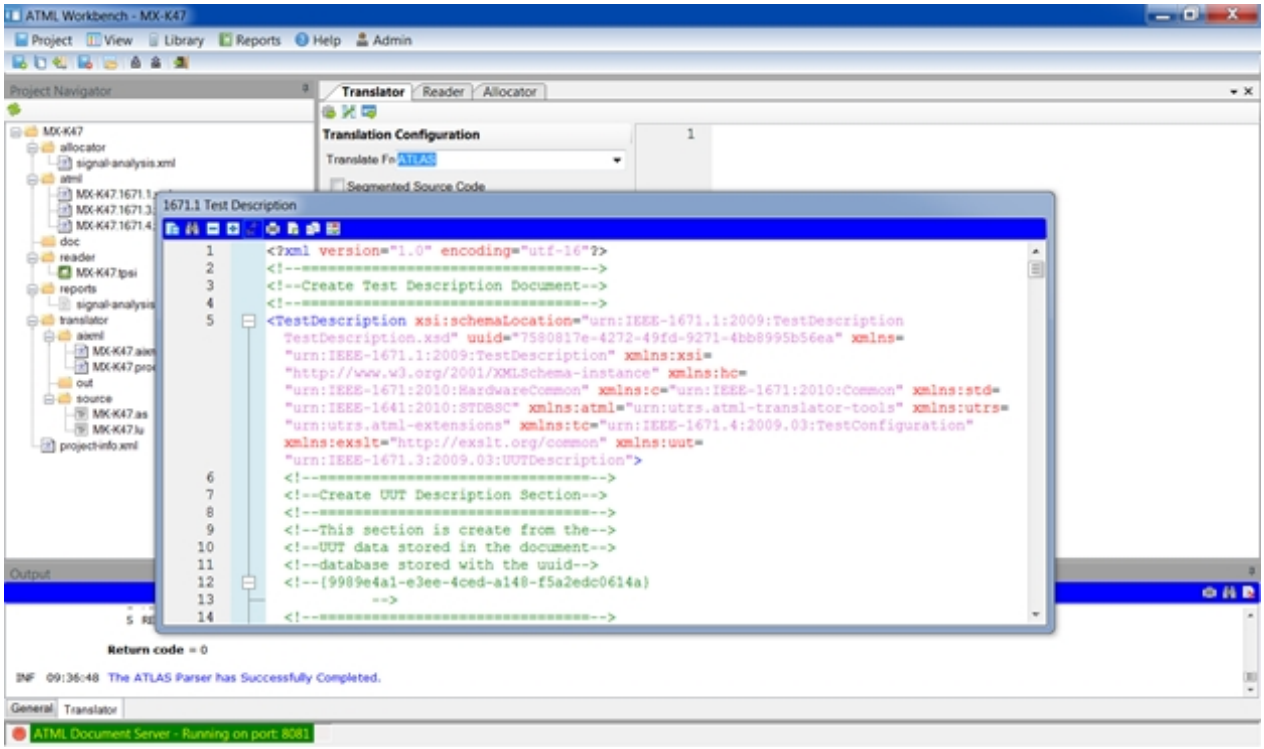
DBG	Debug Messages (Log File Only)	N/A
TRC	Trace Messages	Black
INF	Informational Messages	Blue
WRN	Warning Messages	Brown
ERR	Error Messages	Red

Document/Form Windows

Document/Form Windows

Document and windows can be docked in various areas within the application. The Translator tool, as shown in the first image below, defaults to docking the 1671.1 Test Description document in the lower right quadrant of the application. These windows may be un-docked and re-docked in other locations or kept free floating, if desired, simply by clicking on the window header and dragging the window to the desired location, as shown in the second image below.


ATML Workbench - MX-K47
Project
View
Library
Reports
Help
Admin
Project Navigator
MX-K47
allocator
signal-analysis.xml
atml
MX-K47.1671.1.xml
MX-K47.1671.3.xml
MX-K47.1671.4.xml
doc
reader
MX-K47.tpsi
reports
signal-analysis.html
translator
aixml
MX-K47.aixml.xml
MX-K47.proc_hier.aixml.xml
out
source
MK-K47.as
MK-K47.lu
project-info.xml
Translator
Reader
Allocator
Translation Configuration
Translate Fn ATLAS
Segmented Source Code
Primar FileName
MK-K47.as
MK-K47.lu
Property
Value
Station Type
CASS
Project Name
MX-K47
ID
646a1aa3-f302-405b-9f9c-ae04a5a...
UUT
HYBRID COMMON ID SELF TEST
UUT ID
3f9386e7-8b59-4aad-8735-5ae50c...
Output
5 REQUIRE
Return code = 0
INF 09:36:48 The ATLAS Parser has Successfully Completed.
General
Translator
1671.1 Test Description
1 <?xml version="1.0" encoding="utf-16"?>
2 <!--==========-->
3 <!--Create Test Description Document-->
4 <!--==========-->
5 <TestDescription xsi:schemaLocation=
"urn:IEEE-1671.1:2009:TestDescription
ATML Document Server - Running on port: 8081


ATML Workbench - MX-K47
1671.1 Test Description
1 <?xml version="1.0" encoding="utf-16"?>
2 <!--==========-->
3 <!--Create Test Description Document-->
4 <!--==========-->
5 <TestDescription xsi:schemaLocation="urn:IEEE-1671.1:2009:TestDescription
TestDescription.xsd" uuid="7580817e-4272-49fd-9271-4bb8995b56ea" xmlns=
"urn:IEEE-1671.1:2009:TestDescription" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:hc=
"urn:IEEE-1671:2010:HardwareCommon" xmlns:c="urn:IEEE-1671:2010:Common" xmlns:std=
"urn:IEEE-1641:2010:STDBSC" xmlns:atml="urn:utrs.atml-translator-tools" xmlns:utrs=
"urn:utrs.atml-extensions" xmlns:tc="urn:IEEE-1671.4:2009.03:TestConfiguration"
xmlns:exslt="http://exslt.org/common" xmlns:uut=
"urn:IEEE-1671.3:2009.03:UUTDescription">
6 <!--==========-->
7 <!--Create UUT Description Section-->
8 <!--==========-->
9 <!--This section is create from the-->
10 <!--UUT data stored in the document-->
11 <!--database stored with the uuid-->
12 <!--{9989e4a1-e3ee-4ced-a148-f5a2edc0614a)
13 -->
14 <!--==========-->
Return code = 0
INF 09:36:48 The ATLAS Parser has Successfully Completed.
ATML Document Server - Running on port: 8081

Libraries

Libraries

The ATML Workbench includes a number of libraries for user generated content.

Document Library

Document Library

Provides access to all documents stored in the document database. Documents may be filtered based on document type.

Instrument Library

Instrument Library

Provides access to all the Instrument Descriptions stored in the document database.

Signal Model Library

Signal Model Library

Provides access to all the TSF libraries and signal models stored in the database. Users may also review the signal model description and associated interface properties for specific signal models.

Test Adapter Library

Test Adapter Library

Provides access to all Test Adapter Descriptions stored in the document database.

Test Station Library

Test Station Library

Provides access to all Test Station Descriptions stored in the document database.

UUT Library

UUT Library

Provides access to all UUT Descriptions stored in the document database.

Translator Tool

Translator Tool

The Translator Tool can be used to translate ATLAS source code to an ATML 1671.1 Test Description. The Translator uses a modified compiler train to parse ATLAS source code and generate "ATLAS Intermediate XML" (AIXML). AIXML includes tags to support further translation to ATML or another language. The AIXML is then converted to a 1671.1 Test Description through an XSLT process. The style sheets that define the process may be accessed through the Document Library.

Procedures for the use of the Translator are provided in the Create a 1671.1 Test Description portion of this Help.

Source Code

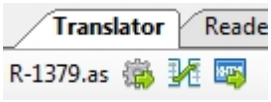
ATLAS source code can be added to the /translator/source folder by selecting the folder, right clicking, and selecting "Add Files" from the pop up menu. This version of the Translator is restricted to ATLAS source code for the Navy's CASS station. All .as and .lu files for a given TPS must be added to the / translator/source folder prior to the start of translation.

The Translator will automatically populate the .as files into the Translation Configuration file list. Source files can be added or removed from the Translation Configuration using the buttons to the right of the Translation Configuration file list.

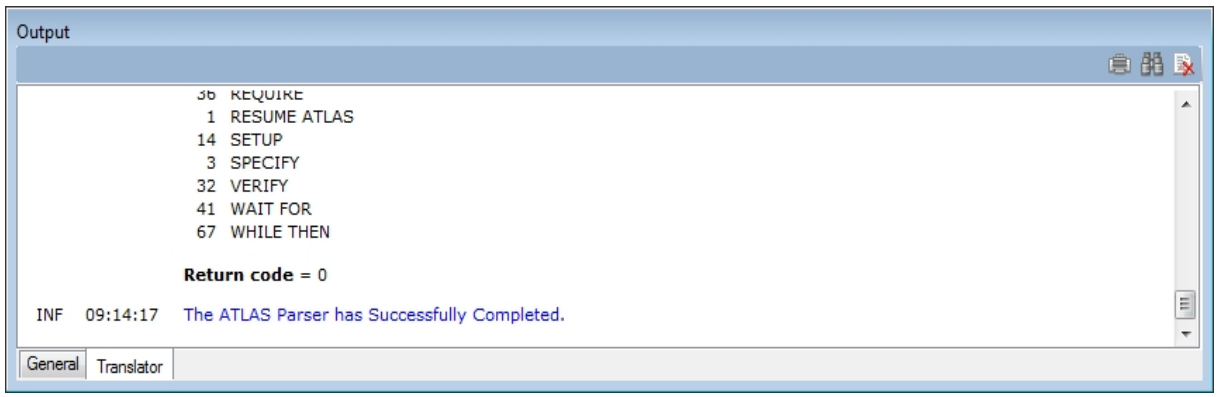
If the source code is segmented, check the "Segmented Source Code" box and identify the primary file by checking the box to the left. The primary file will automatically move to the top of the list.

Translating ATLAS

Once the source files have been added and the translation configuration completed (see Add Source Documents) you can now begin the translation.

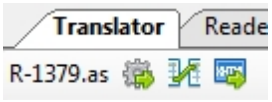
1. Click on the Parse Source Document button from the Translator tool bar 

2. The Translator tab will appear in the Output Window at the lower left corner of the screen to provide ongoing updates as the parsing proceeds.

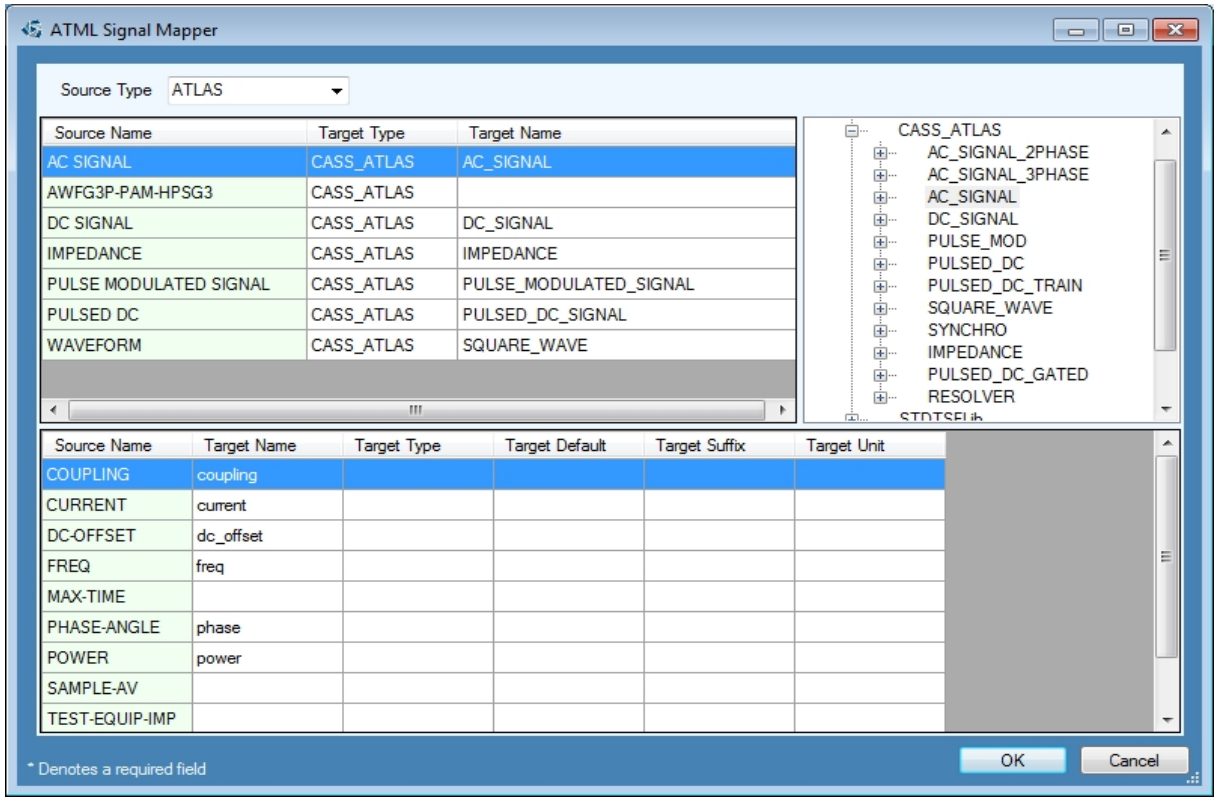


3. Once completed, the focus will shift to the AIXML file opened in its' own document tab. The file will have the name of the project concatenated with .aixml.xml.

4. Once the source files have been translated to AIXML (see Run the AIXML Generator) you can run the ATML Signal Mapper. Ensure the focus is on the Translator Tab.

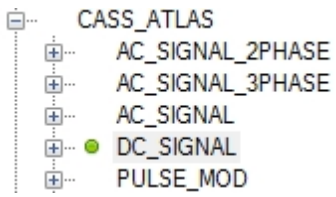
5. Click the Build Signal Map button  available on the Translator tool bar 

6. The ATML Signal Mapper screen will open.



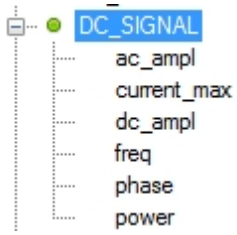
7. The Signal Mapper is divided into three sections. The top left section shows the source signal name and the selected target signal names. Directly under the source signal section is the source signal attribute sections. This section shows the attributes for the selected source signal as well as the selected target signal attributes. Directly to the right of the source signal section is the TSF Library tree which contains a list of all the available TSFs in the Signal Model Library.

8. As source signals are selected, the application will search the TSF tree for the best match based on the source signal attributes and mark that TSF with a green dot.



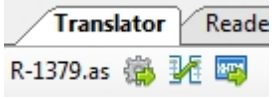
9. To map a TSF signal to a source signal simply select the TSF signal and drag it onto the source signal Target Name of the signal you want to map. The application will then attempt to map the correct signal attributes based on the attribute names.

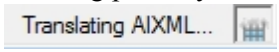
10. Because names of attributes may be different between the TSF and the source signal attributes, you may have to directly map the attributes. To do this open the TSF signal name by clicking the + next to the selected signal name.



11. You can now highlight the TSF attribute name and drag and drop the attribute onto the source signal attribute Target Name of the source signal attribute you would like to map.

12. A 100% mapping is desired but not necessary, however the more data available the better the translation and analysis.

13. Once the source signals have been mapped (see Map Signals to TSF Libraries) you can now run the ATML Generator. Click the AIXML to ATML Translation button  available on the Translator tool bar 

14. The translation process will begin and informational updates will be provided in the Output Window. Because some translations may take a long time to complete, the translation will be performed in the background allowing you to proceed with other work within the ATML Workbench. It should be noted though that during translation some functionality will be disabled in order to permit the current translation to proceed. You will know a translation is taking place by an animation shown on the far right of the status bar at the bottom of the screen 

Source Comparison

ATML Source Comparison Tool

The ATML Source Comparer allows you to trace from the ATML 1671.1 Test Description signal requirement back to the matching ATLAS source code. This allows users to easily confirm that the ATML representation aligns with the original test requirement.

1671.1 Test Description

1671.1 Test Description

The 1671.1 Test Description generated by Version 1.5 of the ATML Workbench is based on the IEEE 1671.1-2009 Test Description standard and includes several extensions created by UTRS to allow the correct representation of the original test procedure program flow. Most terms within CASS ATLAS will appear properly tagged and implemented in AIXML. Selected CASS ATLAS terms are properly tagged and implemented in the ATML generated by the Translator. Some ATLAS terms (e.g., CALCULATE and GOTO) do not have analogs in ATML and cannot be implemented without changes to the standard. ATLAS terms that are not implemented in the ATML Workbench version 1.5 have been assigned to OperationOther.

Reader Tool

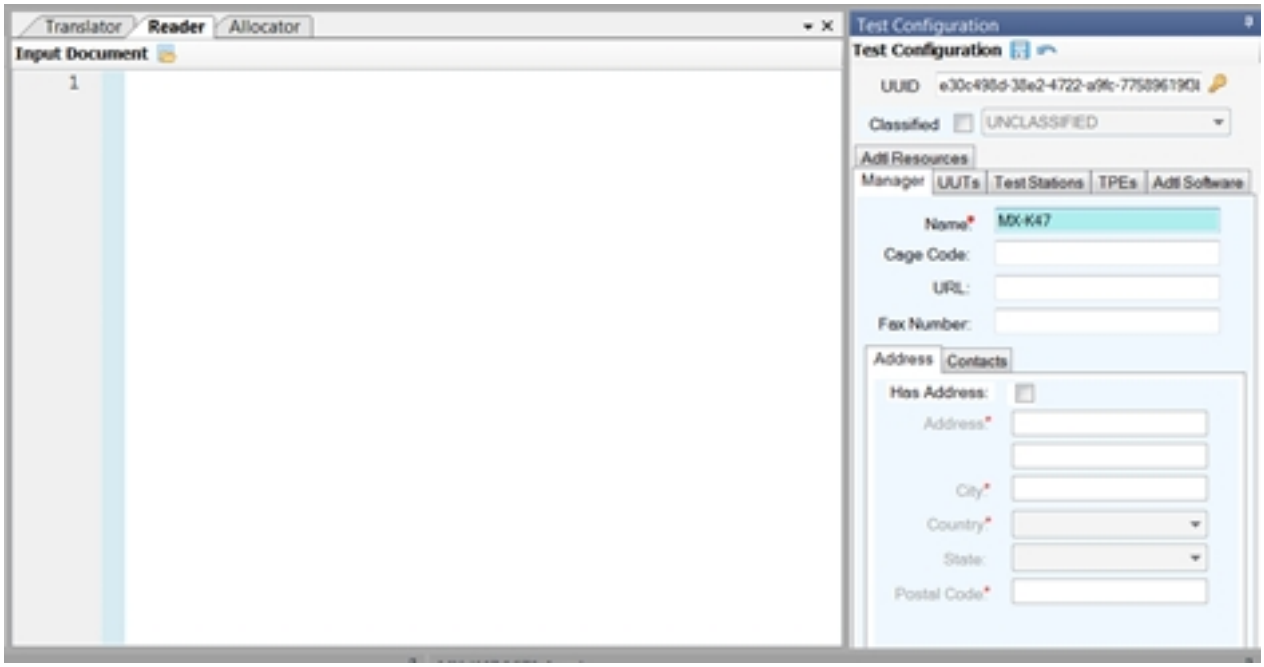
Reader Tool

The Reader Tool can be used to create the 1671.4 Test Configuration file. Procedures for doing so are presented in Create a 1671.4 Test Configuration section of this Help.

Test Configuration Form

Test Configuration Form

The Test Configuration Form appears to the right of the main window when the Reader tab is selected.



The required fields for a test configuration compliant with the 1671.4 schema are marked with red asterisks. Note that Manufacturer Address information is only required if the "Has Address" check box is checked.

Changes to the test configuration can be saved or undone using the buttons in the Test Configuration toolbar. Once saved, changes are automatically placed into the 1671.4 Test Configuration file.

If a .tpsi file is available, the Reader will automatically populate the appropriate database fields during parsing. Otherwise, users may populate the fields using the manual process described here.

The Test Configuration Form includes links to the documents describing related ATE. These links allow users to view, edit, create, or delete the other documents, as desired. Please see Data Entry Forms for more information.

Allocator Tool

Allocator Tool

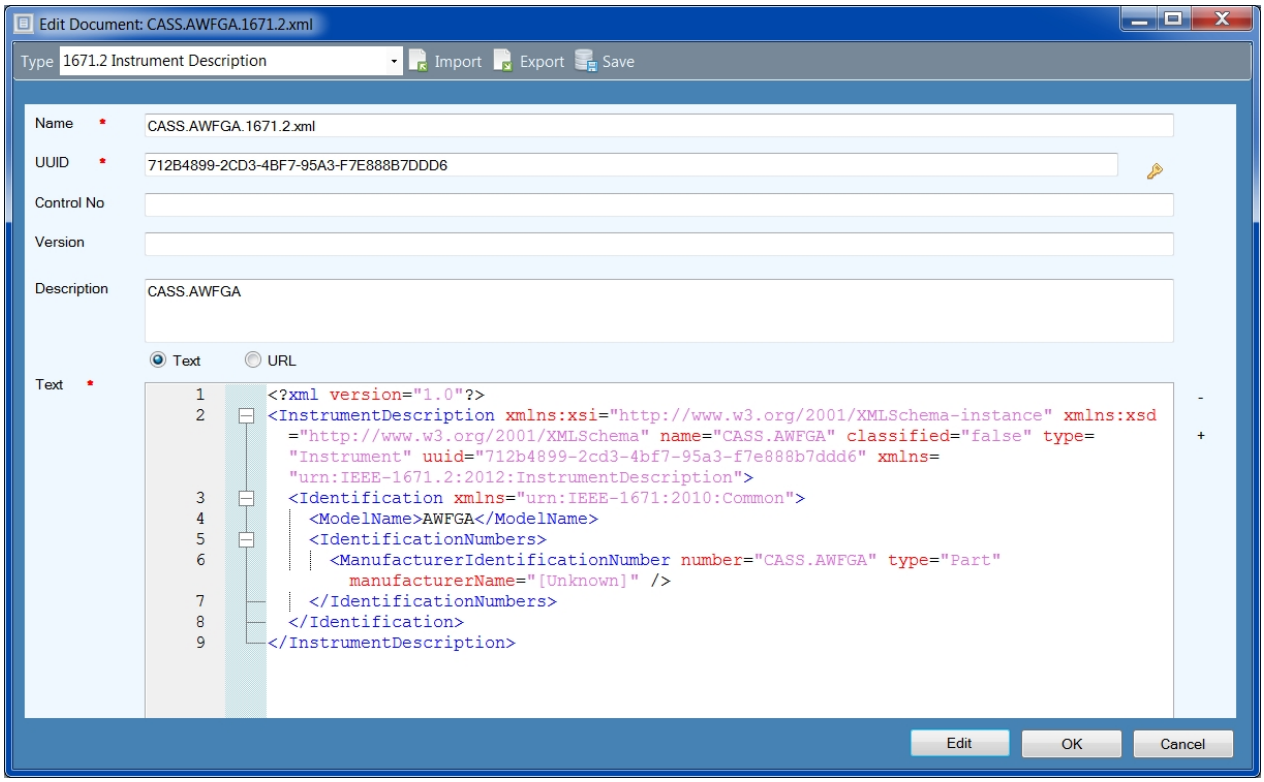
The Allocator Tool can be used to compare the UUT test requirements as described in the 1671.1 Test Description and the 1671.4 Test Configuration files to instrument capabilities described in the ATML Workbench database. Instructions for how to perform this analysis are presented here.

The Allocator matches signals based on name, initially, with a second review comparing interface properties between the signal required by the Test Description and the signal capabilities documented in the instrument database. The ATML Workbench version 1.5 focuses the analysis at the instrument level, as opposed to the test station level or the capabilities of active test adapters. It should be noted that the data entry forms in this version of the ATML Workbench do allow users to document capabilities at the test station or test adapter levels.

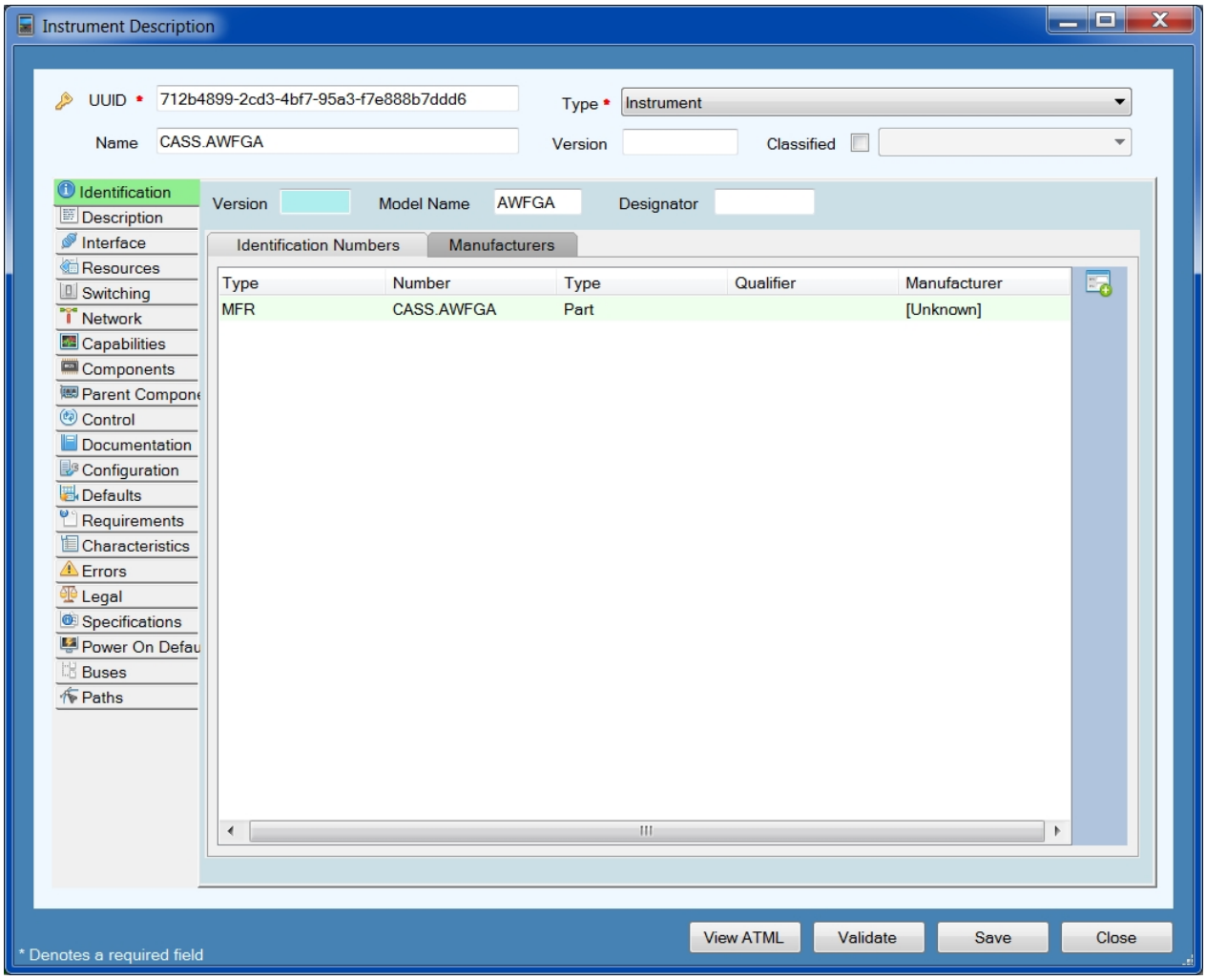
Data Entry

Data Entry Forms

The ATML Workbench includes several ways to manually enter data in the database. Selecting a document from the Document Library will open a pop-up window that allows users to edit the general information for the ATE as well as the XML as shown in the image below.



Clicking on the Edit button at the lower right of the window will open a data entry form that allows users to more easily generate the XML.

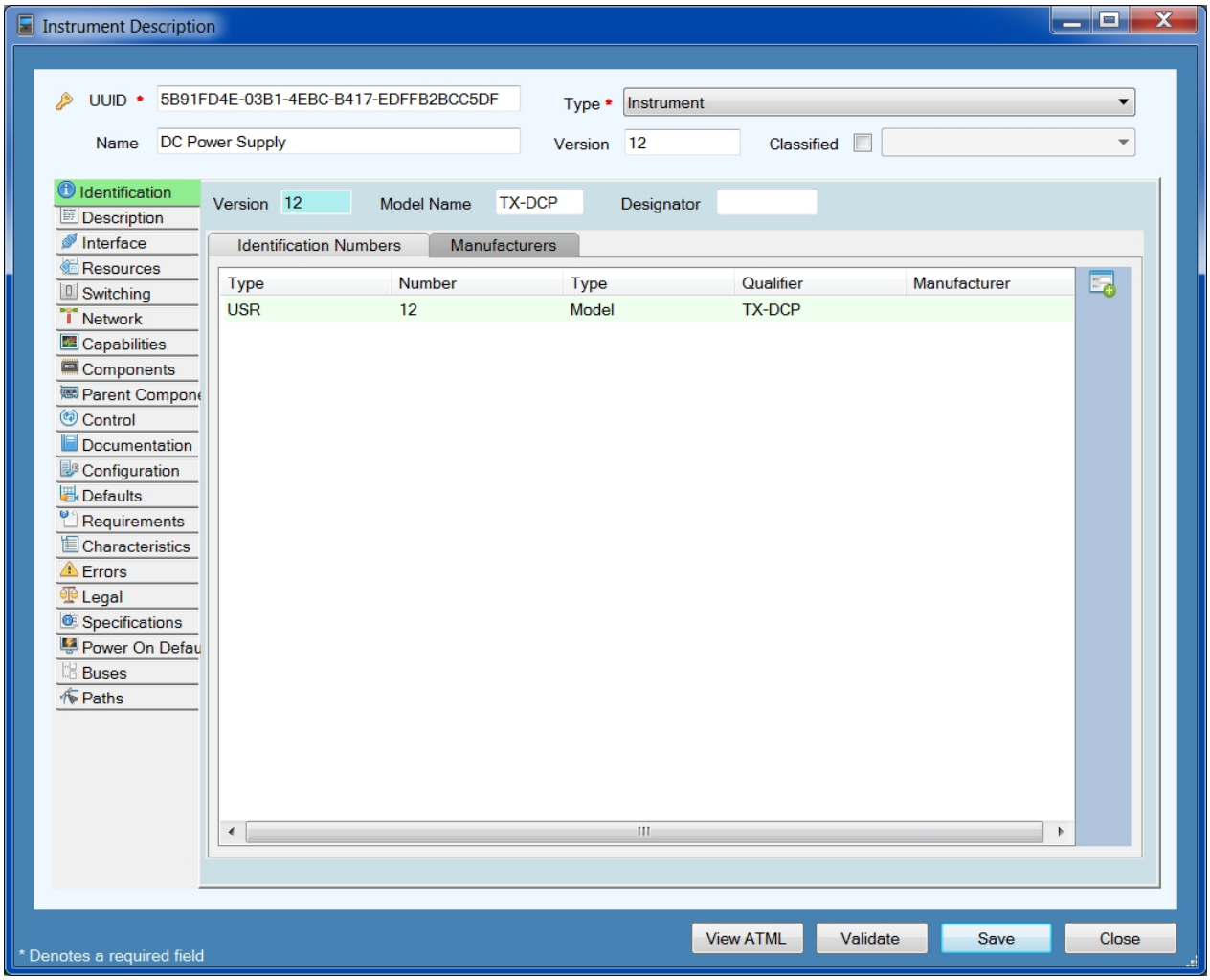


These forms mark the required data for each schema with a red asterisk. Note that some required fields may be on other tabs. The required data for each document type is listed in the appropriate topic within this section of the Help (e.g., for the data required to complete a 1671.6 Test Station Description, see Test Station Description).

1671.2 Instrument Description

Instrument Description

The Instrument Description form provides the means to describe an instrument in its entirety. The layout of the form logically follows the layout of the ATML 1671.2 schema for an instrument description. The order of buttons attempts to mimic the order of operation in the creation and data entry of an instrument description. For example, when defining the capabilities, the interface, resources and switching must be defined first.



UUID (Required)

Pattern: [A-Fa-f0-9]{32}|(\{|\()?[A-Fa-f0-9]{8}-([A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}(\}|\))?
This type is used by other XML schema attributes or elements that will hold a universal unique identifier (UUID), commonly known as either a globally unique identifier (GUID) or UUID. The regular expression defined limits the contents of an attribute to either a single 32-digit hexadecimal string or a 32-digit hex string patterned as [8]-[4]-[4]-[4]-[12] digits.

Type (Required)

Used to describe the type of instrument. It may be one of the following values:

Module
Instrument
Option

Name

A descriptive or common name for the described item.

Version

A string designating the version of the described item.

Classified
The checkbox allows users to indicate that an instrument description is classified and to indicate the level of classification using the drop-down box.

Identification

Description

Interface **(Required)**
One interface (either a Connector or a Port) is required.

Resources

Switching

Network

Capabilities

Components

Parent Components

Control

Documentation

Configuration

Defaults

Requirements

Characteristics

Errors

Legal

Specifications

Power On Defaults

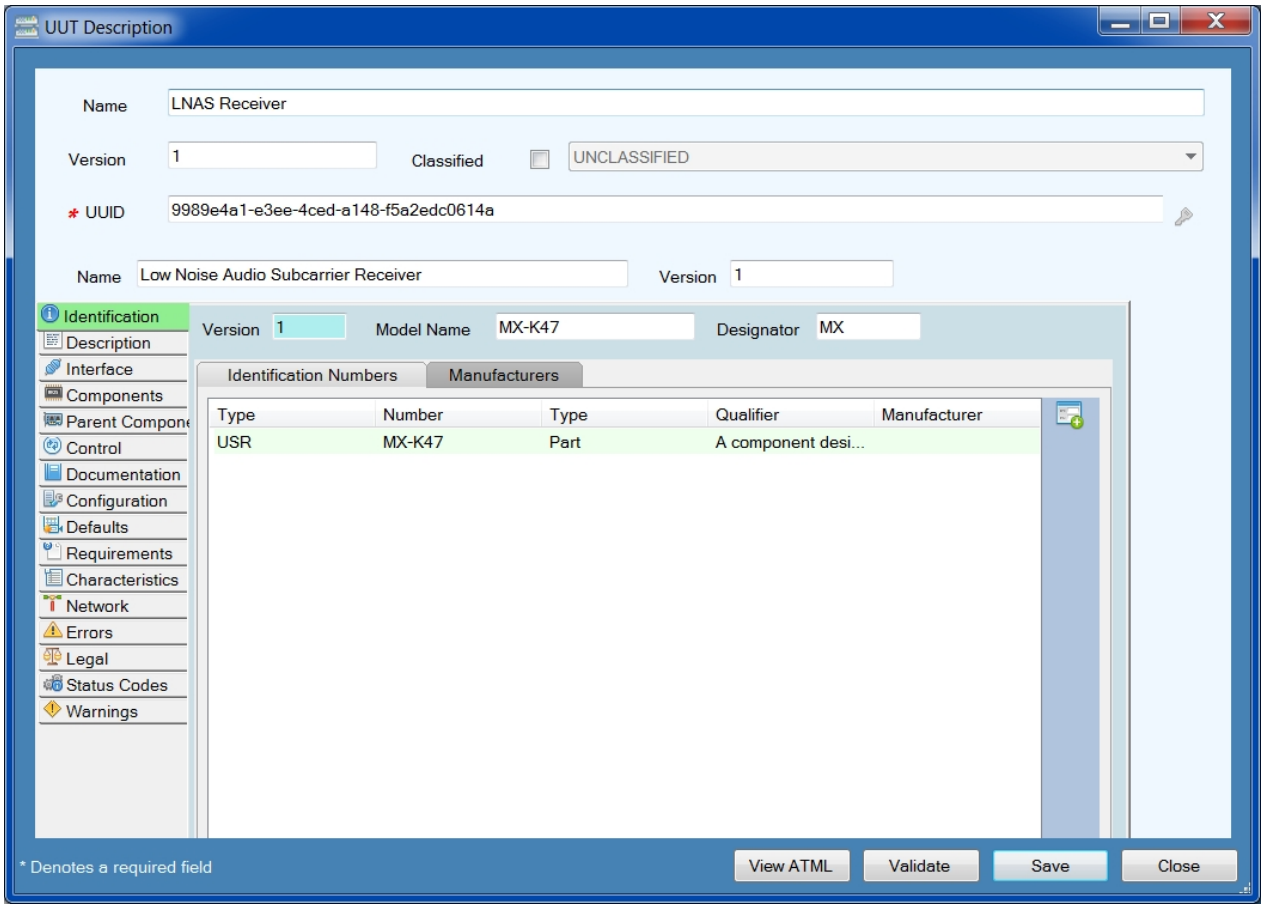
Buses

Paths

1671.3 UUT Description

UUT Description

The 1671.3 UUT Description documents the characteristics of the UUT. A minimal UUT Description is required for the creation of a new project.



Name **(Required)**
A descriptive or common name for the described item.

Version
A string designating the version of the described item.

Classified
The checkbox allows users to indicate that an instrument description is classified and to indicate the level of classification using the drop-down box.

UUID **(Required)**
Pattern: [A-Fa-f0-9]{32}|(\{|\()?[A-Fa-f0-9]{8}-([A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}(\}|\))?
This type is used by other XML schema attributes or elements that will hold a universal unique identifier (UUID), commonly known as either a globally unique identifier (GUID) or UUID. The regular expression defined limits the contents of an attribute to either a single 32-digit hexadecimal string or a 32-digit hex string patterned as [8]-[4]-[4]-[4]-[12] digits.

Model Name (Required)
A string identifying the model of the item.

Identification

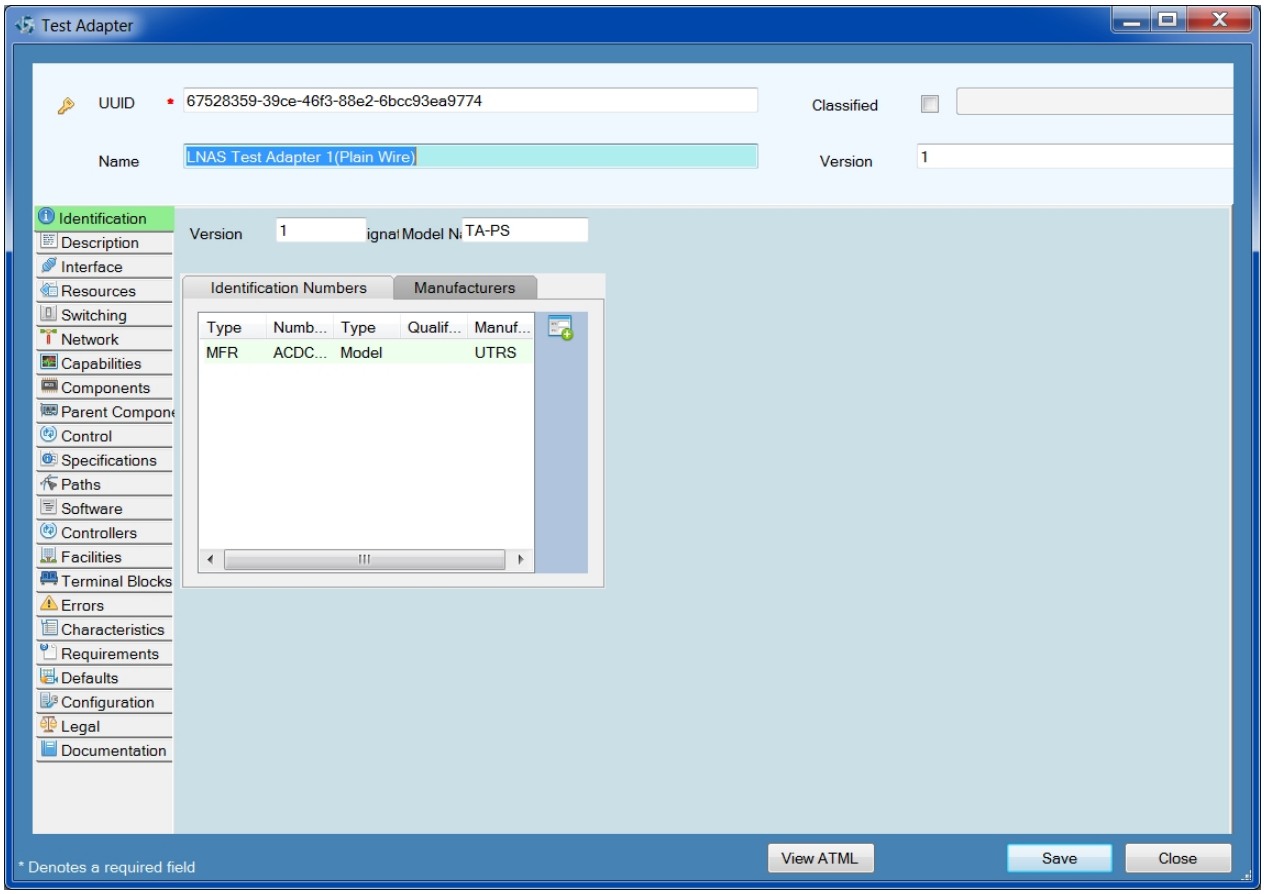
Description

Interface **(Required)**
One interface (either a Connector or a Port) is required.

1671.5 Test Adapter Description

Test Adapter Description

The Test Adapter Description documents the characteristics of the test adapter.



UUID (Required)

Pattern: [A-Fa-f0-9]{32}|(\{|\()?[A-Fa-f0-9]{8}-([A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}(\}|\))?
This type is used by other XML schema attributes or elements that will hold a universal unique identifier (UUID), commonly known as either a globally unique identifier (GUID) or UUID. The regular expression defined limits the contents of an attribute to either a single 32-digit hexadecimal string or a 32-digit hex string patterned as [8]-[4]-[4]-[4]-[12] digits.

Classified

The checkbox allows users to indicate that an instrument description is classified and to indicate the level of classification using the drop-down box.

Name

A descriptive or common name for the described item.

Version

A string designating the version of the described item.

Identification

Description

Interface (Required)

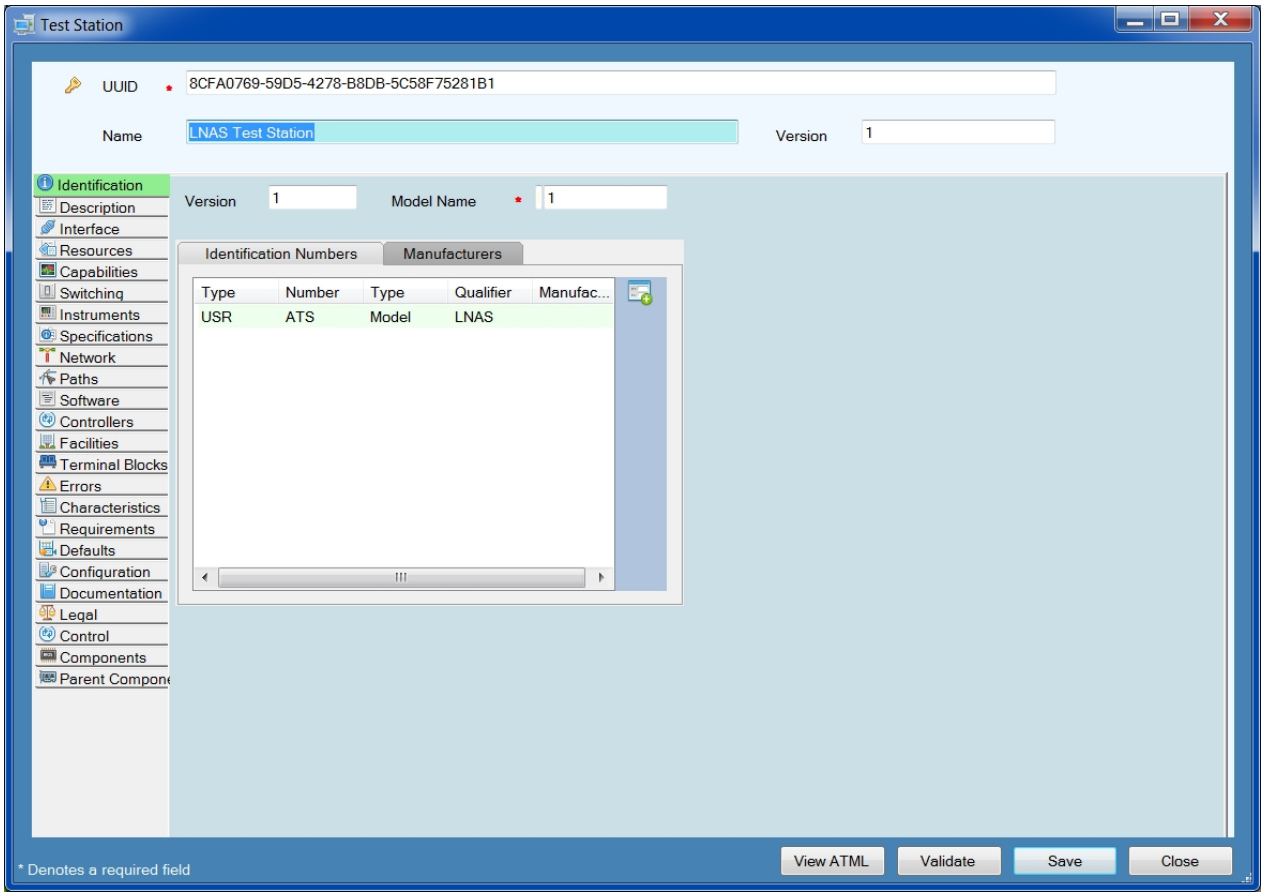
One interface (either a Connector or a Port) is required.

Resources

1671.6 Test Station Description

Test Station Description

The Test Station Description documents the characteristics of a test station.



UUID (Required)
Pattern: [A-Fa-f0-9]{32}|(\{|\()?[A-Fa-f0-9]{8}-([A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}(\}|\))?
This type is used by other XML schema attributes or elements that will hold a universal unique identifier (UUID), commonly known as either a globally unique identifier (GUID) or UUID. The regular expression defined limits the contents of an attribute to either a single 32-digit hexadecimal string or a 32-digit hex string patterned as [8]-[4]-[4]-[4]-[12] digits.

Name
A descriptive or common name for the described item.

Version
A string designating the version of the described item.

Model Name (Required)
Enter the model name of the test station.

Identification

Description

Interface (Required)
One interface (either a Connector or a Port) is required.

Resources

Capabilities

Buses

Buses

This field shall be used to document an item's communication bus.

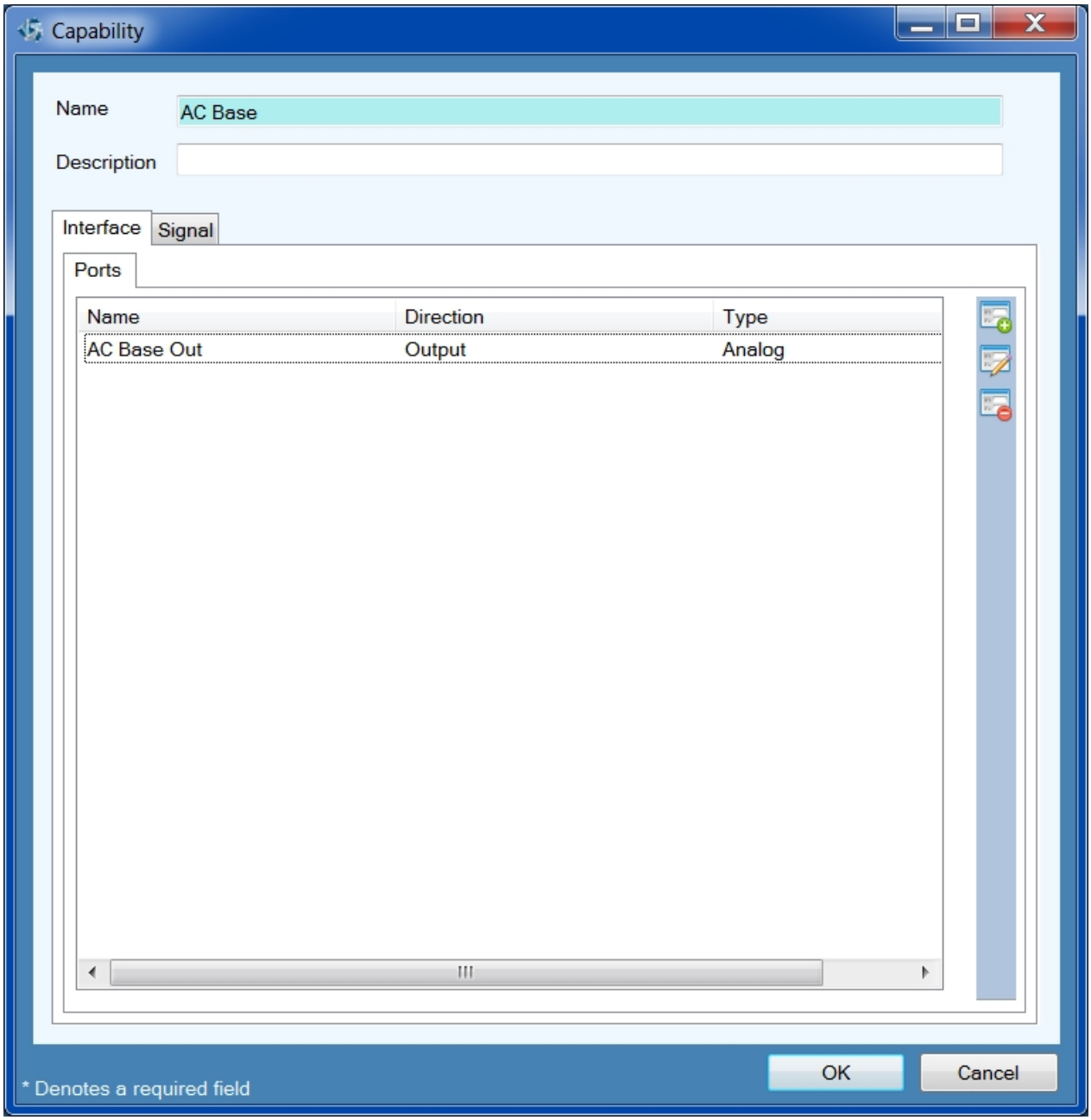
Capabilities

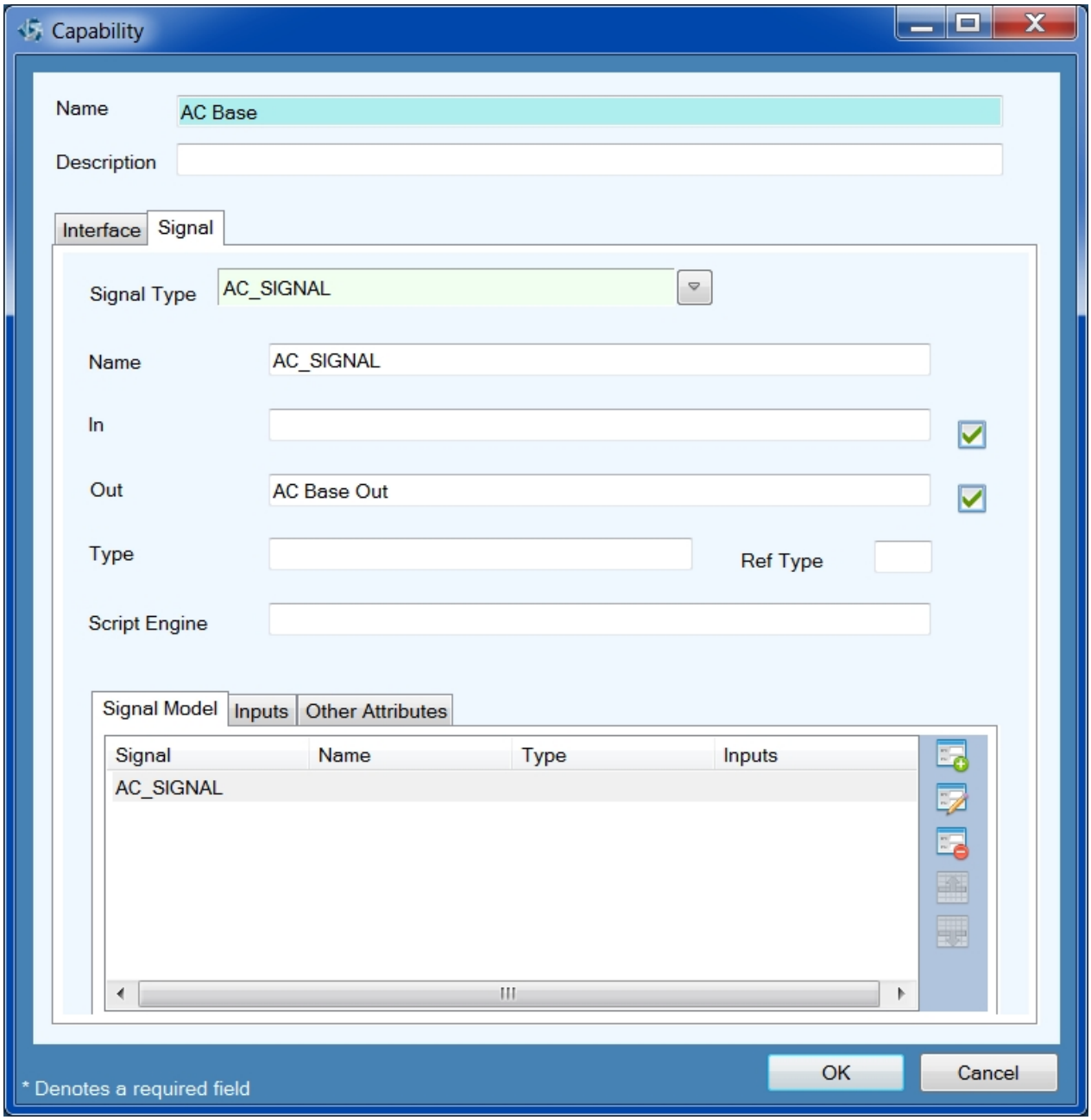
Capabilities

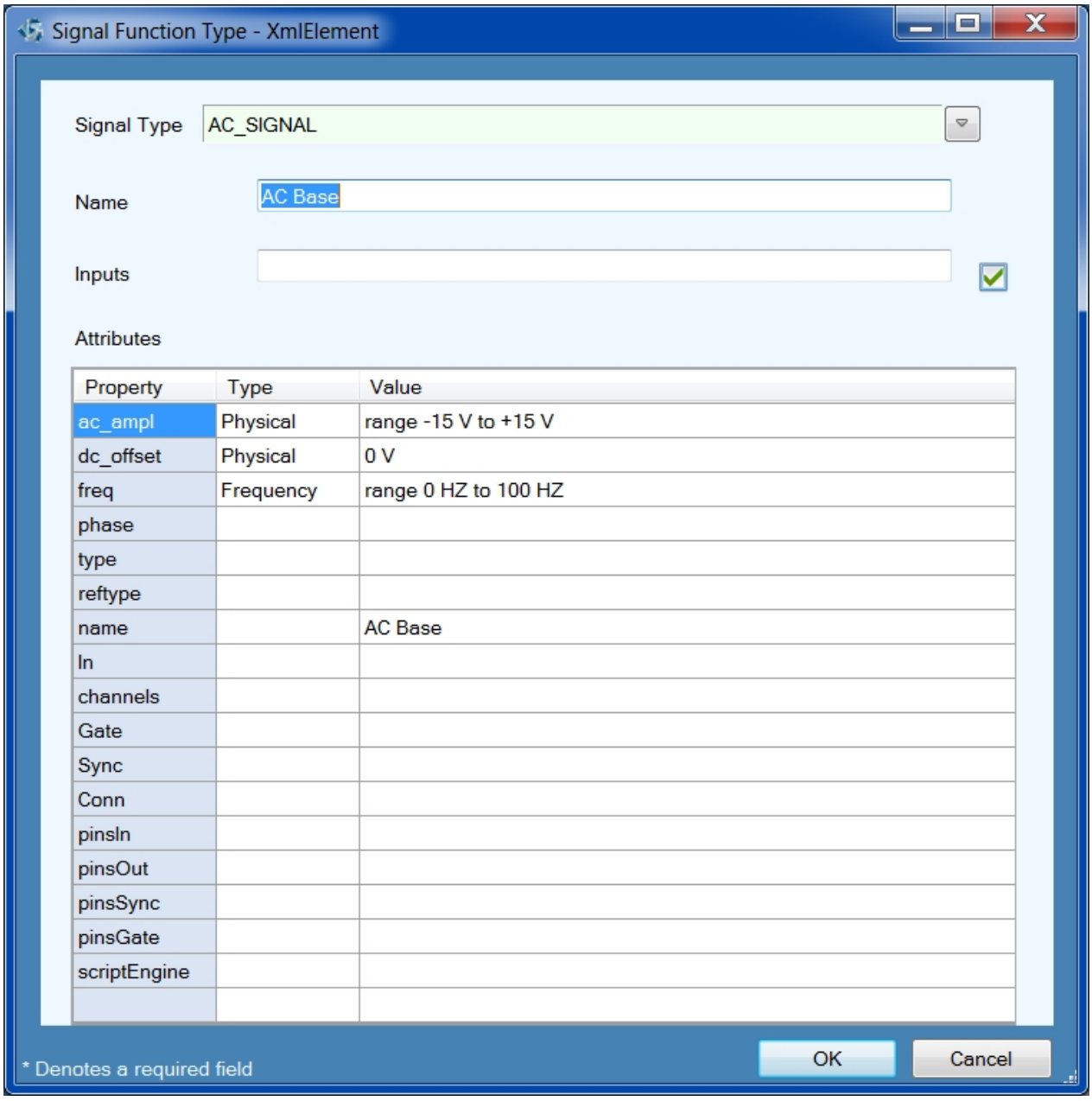
The Capabilities complex type shall be used as the base type for XML schema elements intended to document capabilities and interconnections of a hardware item.

When describing the capabilities of a hardware item the following steps should be taken in the following order:

1. **Interface** - Describe the interface of the hardware item, both physical and logical.
2. **Resources** - Describe the "logical" resources available within the hardware item.
3. **Network** - Create the connections between the resource interfaces and the hardware item interfaces.
4. **Switching** - Specify signal routing options.
5. **Capabilities** - Describe the capability signals available from the hardware item by associating the logical interface ports with signal resources. First create a Port and associate a signal. For instruments, test adapters, and test stations, the signal function attributes should reflect the capability range of the item with respect to that signal. When entering attributes, put the word "range" before a range of values.


Capability
Name
AC Base
Description
Interface
Signal
Ports
Name
Direction
Type
AC Base Out
Output
Analog
* Denotes a required field
OK
Cancel


Capability
Name
AC Base
Description
Interface
Signal
Signal Type
AC_SIGNAL
Name
AC_SIGNAL
In
Out
AC Base Out
Type
Ref Type
Script Engine
Signal Model
Inputs
Other Attributes
Signal
Name
Type
Inputs
AC_SIGNAL
OK
Cancel
* Denotes a required field

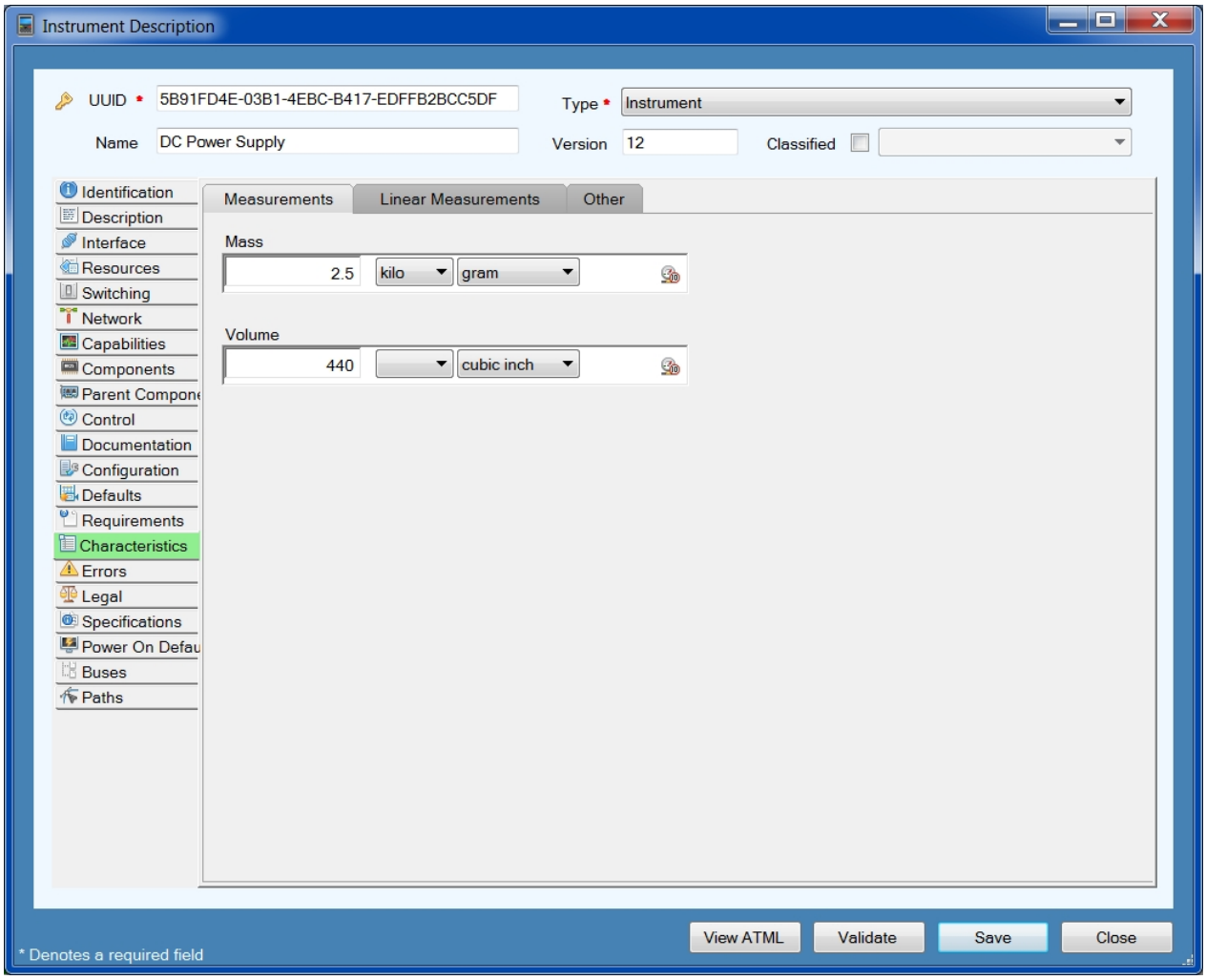


1. **Capabilities Map** - Map the resources and capabilities together by building the path from one to the other.

Characteristics

Characteristics

These fields shall be used to document the physical characteristics of the item.



Components

Components

These fields shall be used to document the physical components which make up the item.

Configuration

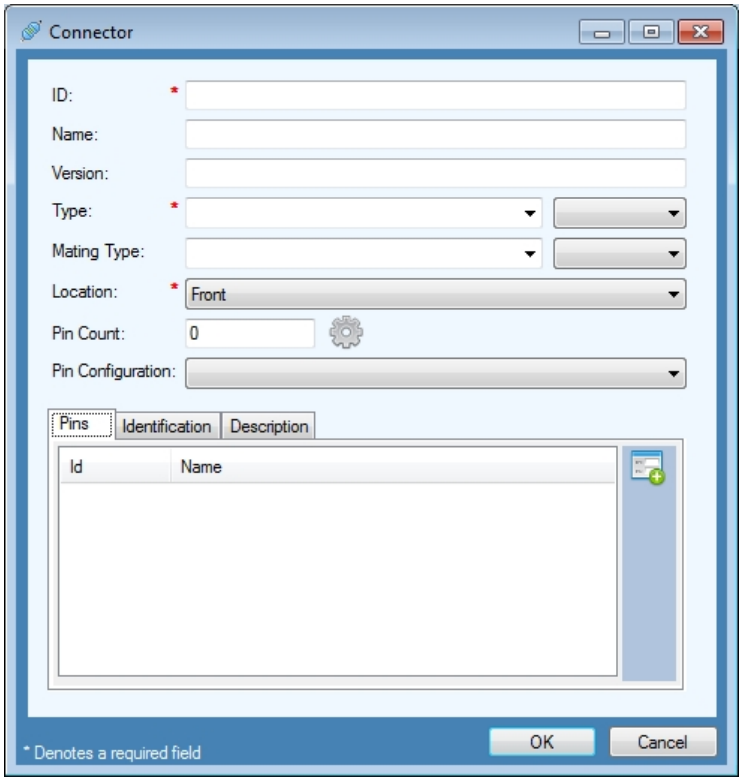
Configuration

These fields shall be used to document the configuration options for the item (i.e., the settings which can be changed by the user but will persist after a power cycle).

Connector

Connector

The PhysicalInterface/Connectors/Connector child element shall identify a physical connector of a hardware item.



ID (Required)
A user-defined string uniquely identifying the connector. Example: J1.

Name
A descriptive or common name for the described item

Version
A string designating the version of the described item.

Type (Required)
A descriptive or common name for the type of connector. Example: MIL-C-38999.

Mating Type
A descriptive or common name for the mating connector. Example: The mating connector for a 15-pin d-shell connector (male) is a 15-pin d-shell connector (female). These fields will automatically populate based on the choices made in the Type field, above.

Location (Required)
A descriptive or common name of where the connector is located. Example: Front Panel.

Pin Count
Number of pins in the connector. By entering a number here and pressing the button, a place holder connector pin entry will be made.

Pin Configuration
Documents the configuration of the pins (e.g., Serial 9 Pin).

Pins
Identifies the pins by ID and name.

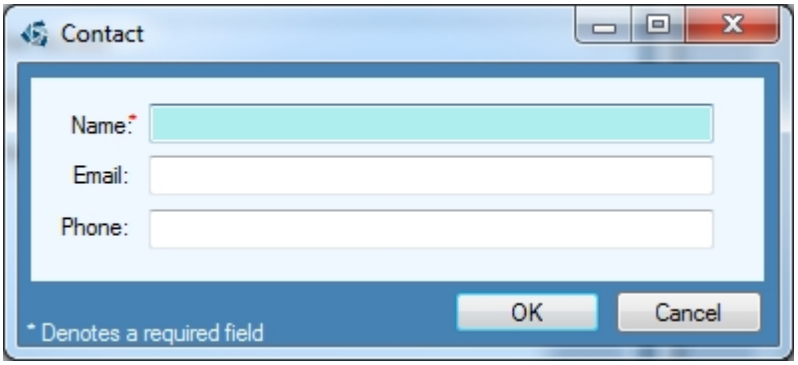
Identification
The ItemDescription/Identification child element shall identify a class of the described item.

Description
The ItemDescription/Description child element shall contain a free-form textual description of the item described.

Contact

Contact

The ManufacturerData/Contacts/Contact child element shall identify the contact's email address, name, and telephone number.



Name (Required)
The contact's given name.

Email
The email address for the contact.

Phone
The contact's telephone number.

Control

Control

Documents the firmware, drivers, control languages, and tools available for the item.

Controllers

Controllers

Documents the properties of a controller item.

Audio Capabilities
Shall identify the audio capabilities of the controller.

Installed Software
Shall identify all software installed on the controller.

Operating Systems (Required)
Shall identify the operating system installed on the controller.

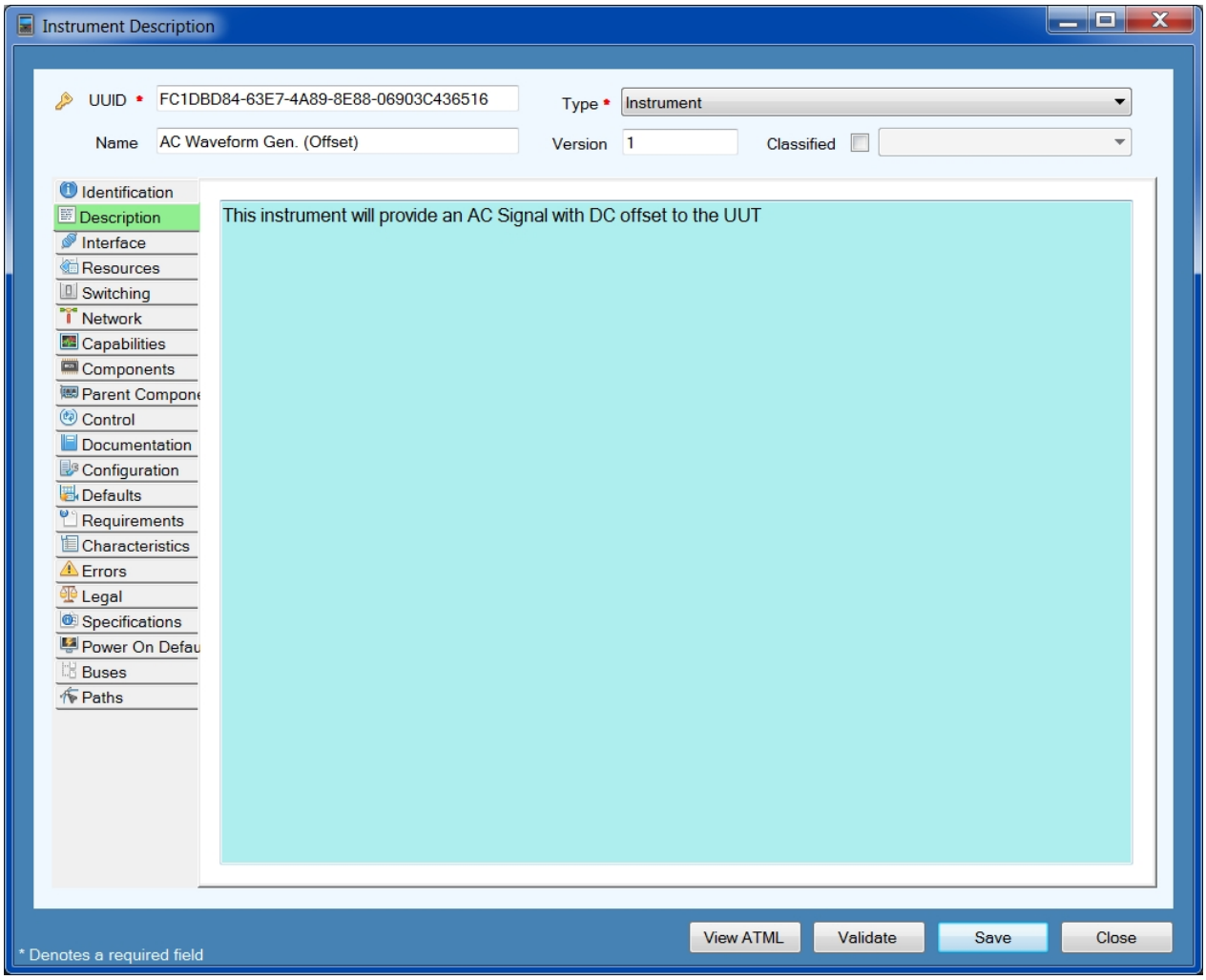
Peripherals
Shall identify the peripherals installed on the controller.

Physical Memory (Required)
Shall identify the physical memory of the controller.

Processor (Required)
Shall identify the controller's processor.

Description

Description
The ItemDescription/Description child element shall contain a free-form text description of the item described.



Defaults

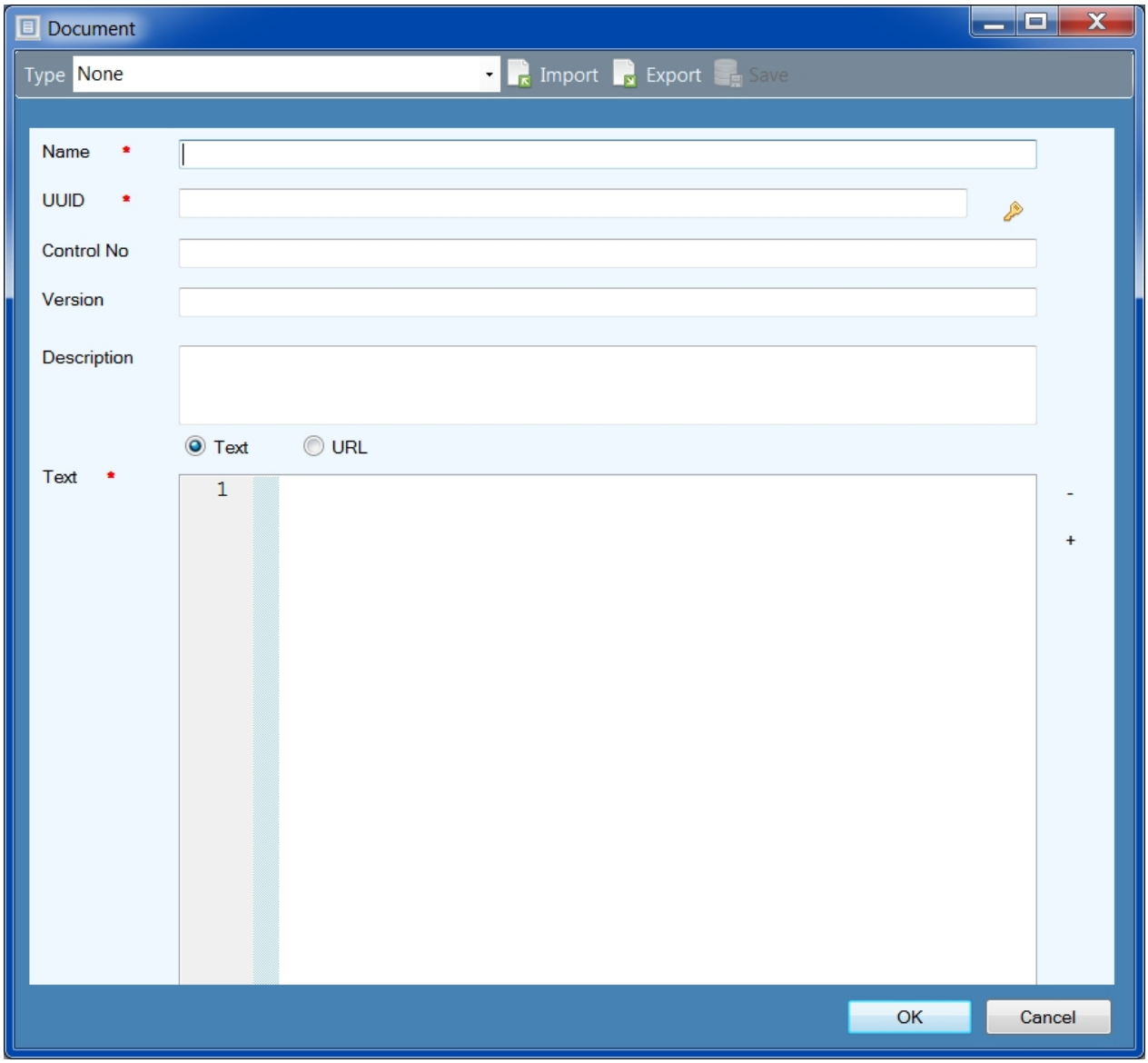
Defaults

Shall identify the default factory settings of the item.

Documentation

Documentation

Shall identify the documentation for the item.



Errors

Errors

Shall document the error codes associated with the item.

Facilities

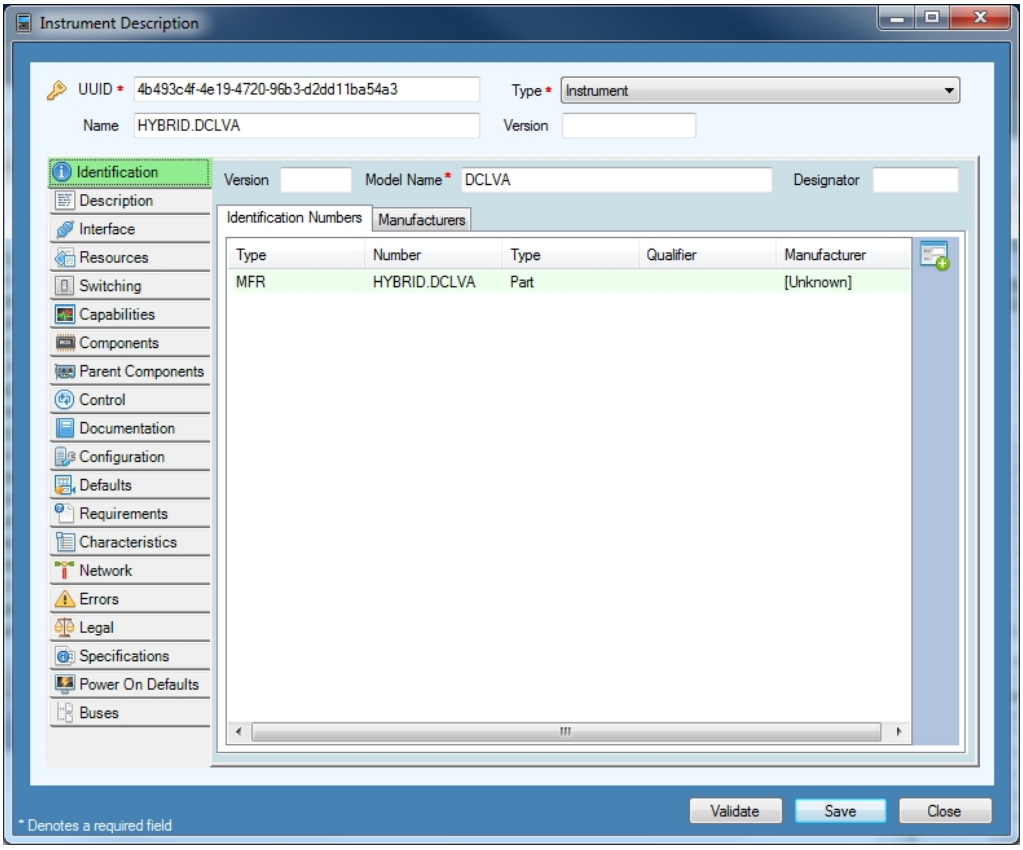
Facilities

Shall document the facility requirements for cooling, hydraulic, and pneumatic supply as well as the facility interface requirements.

Identification

Identification

The Identification form provides the means to add multiple Identification Numbers and Manufacturers to a piece of hardware. When an Identification is used, a Model Name must be provided.



Version

The ItemDescription/Identification/Version child element shall contain a textual description of the version of the item.

Model Name (Required)

The ItemDescription/Identification/ModelName child element shall contain the model name of the item.

Designator

An alphanumeric string that identifies an item within a larger assembly. For example, a reference designator such as A25 to indicate a circuit card number.

Identification Numbers

The ItemDescription/Identification/IdentificationNumbers child element shall be a collector for an unbounded set of Identification Number or ManufacturerIdentificationNumber child elements. This element identifies multiple part or model numbers for the described item (such as a user and/or manufacturer part number).

Manufacturers

The ItemDescription/Identification/Manufacturers child element shall identify the manufacturers of the item.

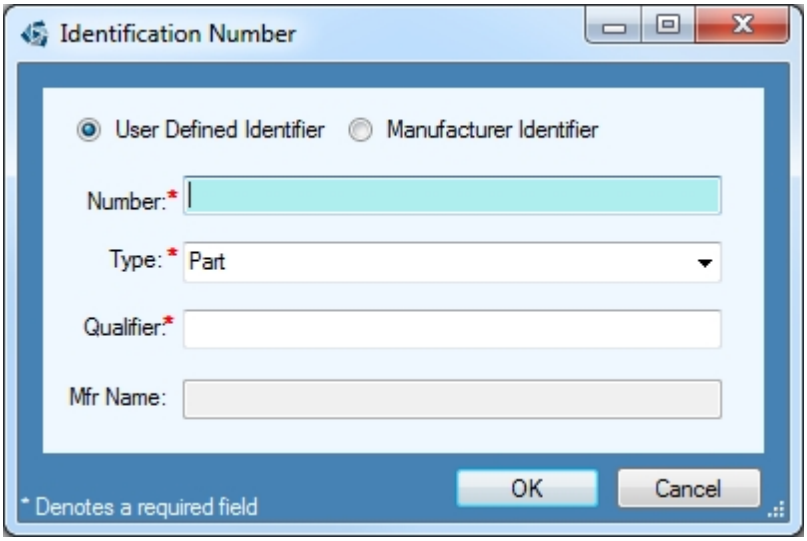
Contacts

The ManufacturerData/Contacts child element shall be a collector for an unbounded set of child ManufacturerData/Contacts/Contact elements.

Identification Number

Identification Number

The ItemDescription/Identification/IdentificationNumbers/IdentificationNumber child element shall provide for multiple end-user-assigned part or model numbers for the described item.



User Defined Identifier

The ItemDescription/Identification/IdentificationNumbers/IdentificationNumber child element shall provide for multiple end-user-assigned part or model numbers for the described item.

Manufacturer Identifier

The ItemDescription/Identification/IdentificationNumbers/ManufacturerIdentificationNumber child element shall provide for multiple manufacturers' assigned part or model numbers, which are not the end-users' assigned part number, for the described item.

Number

The part number of the entity.

Type

An indication of whether this is a part number, model number, or other.

Qualifier

An adjective providing additional descriptive data that further specify or identify the ‘number' attribute. Example: the identification number specified by the user.

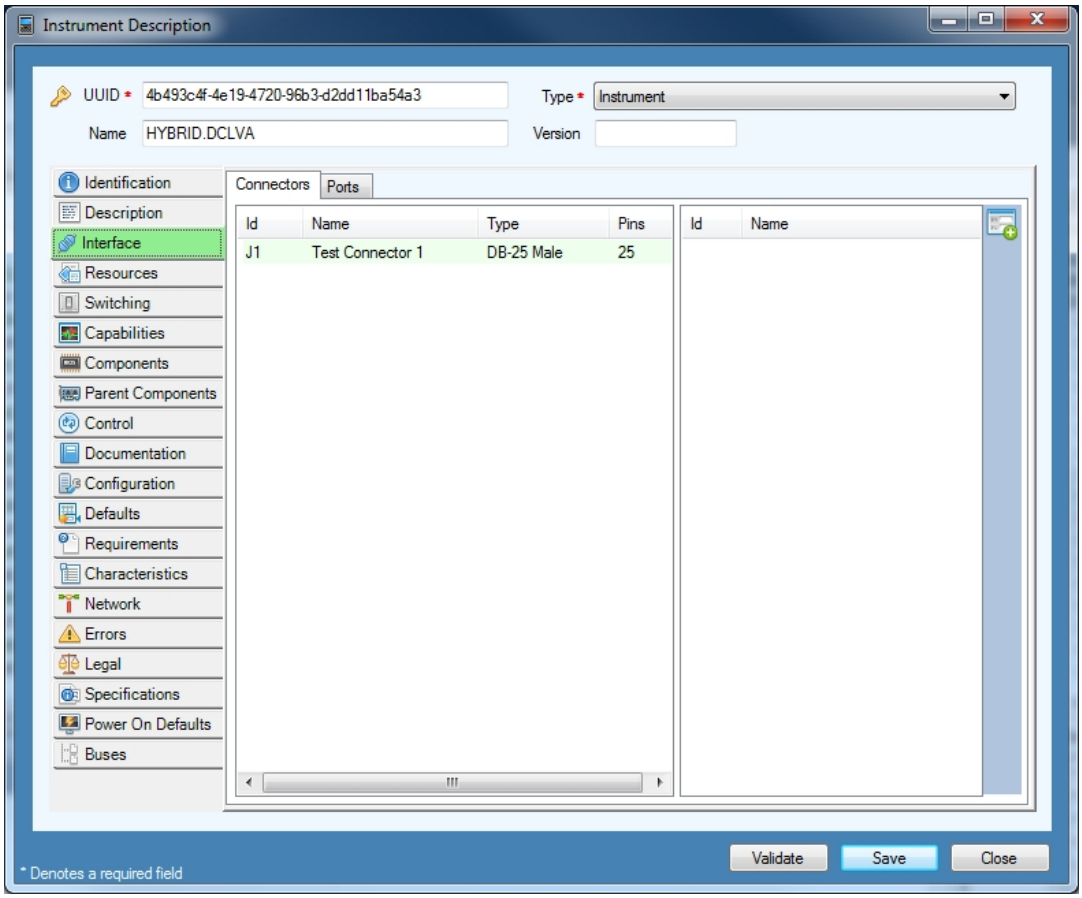
Mfr Name

A descriptive or common name for the manufacturer.

Interface

Interface

The HardwareItemDescription/Interface child element shall identify the electrical interfaces to the hardware item. Typical practice would be to add all of the connector information prior to the port data. This way the connector pin data will be available when defining a port.



Connectors

The PhysicalInterface/Connectors child element shall be a collector for an unbounded set of child PhysicalInterface/Connectors/Connector elements.

Ports

The PhysicalInterface/Ports child element shall be a collector for an unbounded set of child c:Port elements.

Legal

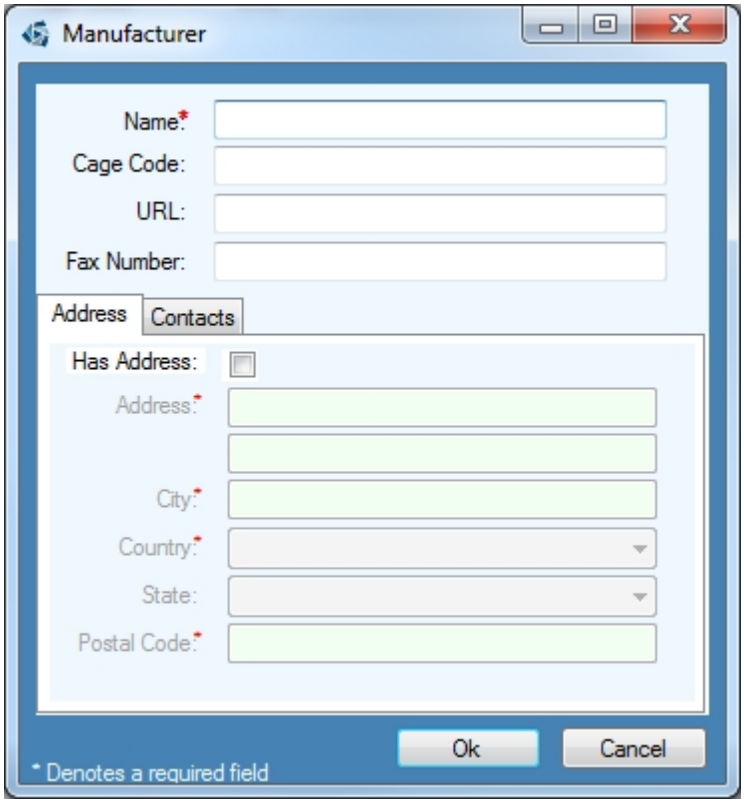
Legal

Shall be a collector element of legal documents for the item. May include conformance, exportability, license, safety, or warranty documentation.

Manufacturer

Manufacturer

The ItemDescription/Identification/Manufacturers/Manufacturer child element shall identify the manufacturer of the item.



Name (Required)

A descriptive or common name for the manufacturer.

Cage Code

The commercial and government entity (CAGE) code for the company indicated by the name attribute.

URL

The ManufacturerData/URL child element shall contain the URL of the Web site for the manufacturer of the items.

Fax Number

The ManufacturerData/FaxNumber child element shall contain a textual representation of the facsimile telephone number of the manufacturer of the item.

Has Address

Check if a manufacturer address is to be provided.

Address 1 (Required if Has Address)

The MailingAddress/Address1 child element shall contain a textual description of the physical street address.

Address 2

The MailingAddress/Address2 child element shall contain a textual description of additional street address information (e.g., suite number, mail stop) that shall be associated with c:MailingAddress/Address1.

City (Required if Has Address)

The MailingAddress/City child element shall contain a textual description of the city that shall be associated with c:MailingAddress/Address1.

Country (Required if Has Address)

The MailingAddress/Country child element shall contain a textual description of the territory occupied by a nation.

State

The MailingAddress/State child element shall contain the U.S. state (Examples: Florida and Hawaii) typically appended to an U.S. postal address.

Postal Code (Required if Has Address)

The MailingAddress/PostalCode child element shall contain a series of letters and/or digits typically appended to the postal address for the purposes of sorting mail (Example: U.S. Postal Service ZIP code).

Network

Network

This field shall be used to document the connections between the resource interfaces and the hardware item interfaces.

Parent Components

Parent Components

These fields shall be used to document the components of which the item is a part.

Paths

Paths

Shall define a signal path within the test equipment.

Path Nodes (Required)

Shall define the beginning and end nodes associated with a single or multiwire path. Switches may be present within a wire path.

Resistance

Shall identify the resistance of the path.

Signal Delays

Shall identify the delay times of the signal through the paths.

SParameters

Shall identify a specific S-parameter associated with the path.

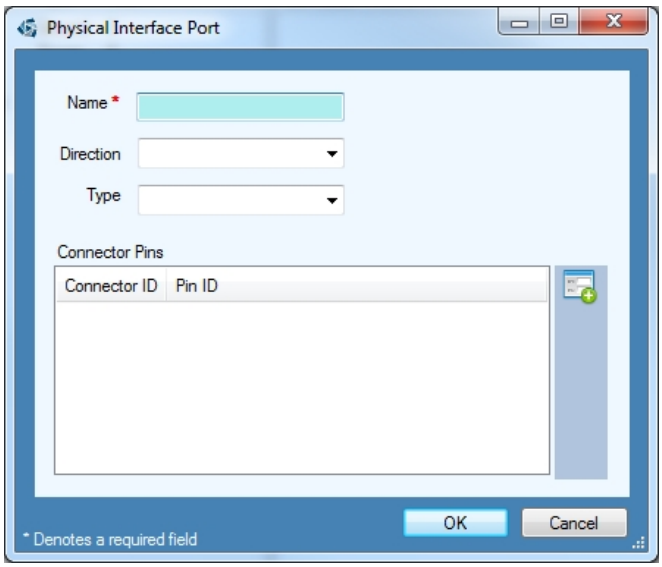
VSWRValues

Shall identify the voltage standing wave ratio(s) associated with a single- or multiwire path.

Port

Port

A Port is a logical representation to discrete inputs or outputs of a physical interface. The connector pins represent physical pins located in a connector(s).



Name

A descriptive or common name for the port.

Direction

An enumeration providing for the specification of the direction in which data moves on the described port. Enumeration values are Input, Output, and Bi-Directional.

Type

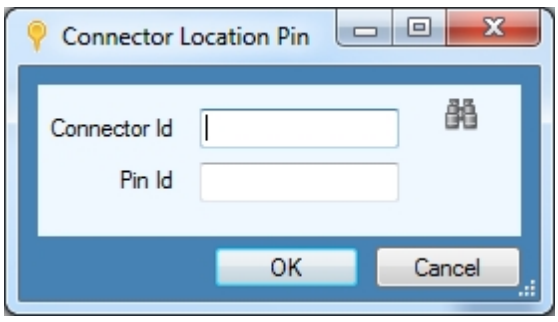
An enumeration providing for the specification of the type of the described port. Enumeration values are Ground, Analog, Digital, Power, Optical, or Software.

Connector Pins

The PhysicalInterface/Ports/ConnectorPins child element shall be a collector for an unbounded set of c:ConnectorPin child elements.

Connector Pin

The PhysicalInterface/Ports/ConnectorPins/ConnectorPin child element shall identify a physical pin of a connector.



Connector ID

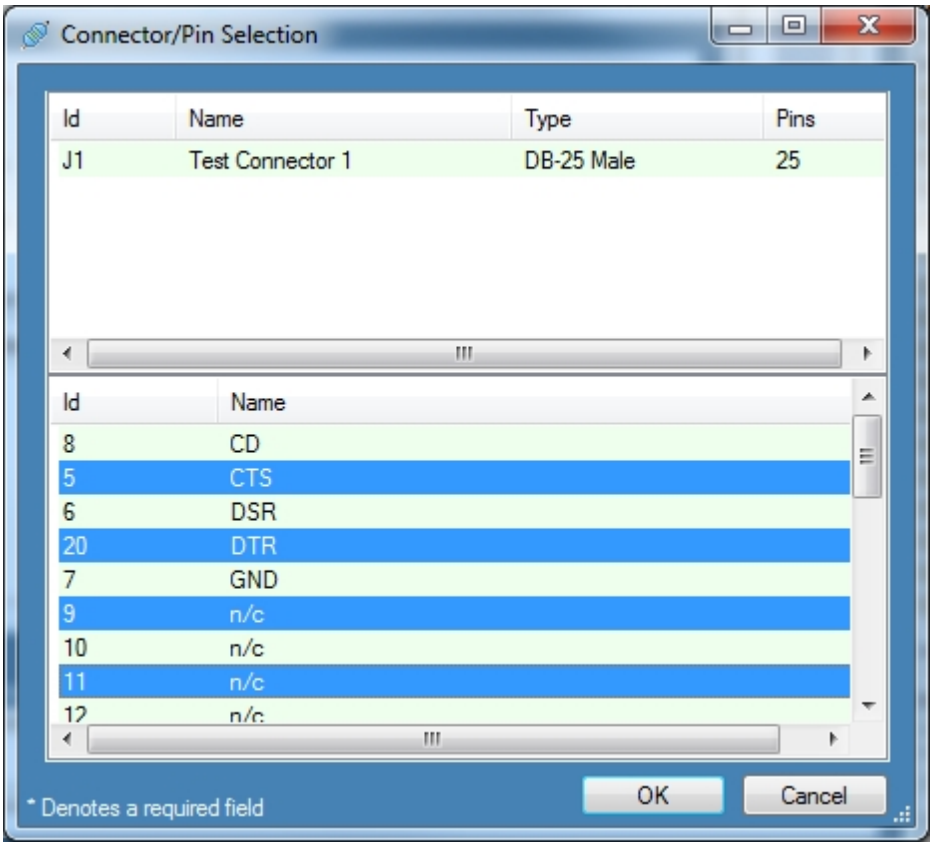
A user-defined string uniquely identifying the connector.

Pin ID

A user-defined string uniquely identifying the pin within the connector.

Connector/Pin Selection

The pins of interest for a specific port definition may be selected by holding down the Ctrl key and selecting multiple pins.



Power On Defaults

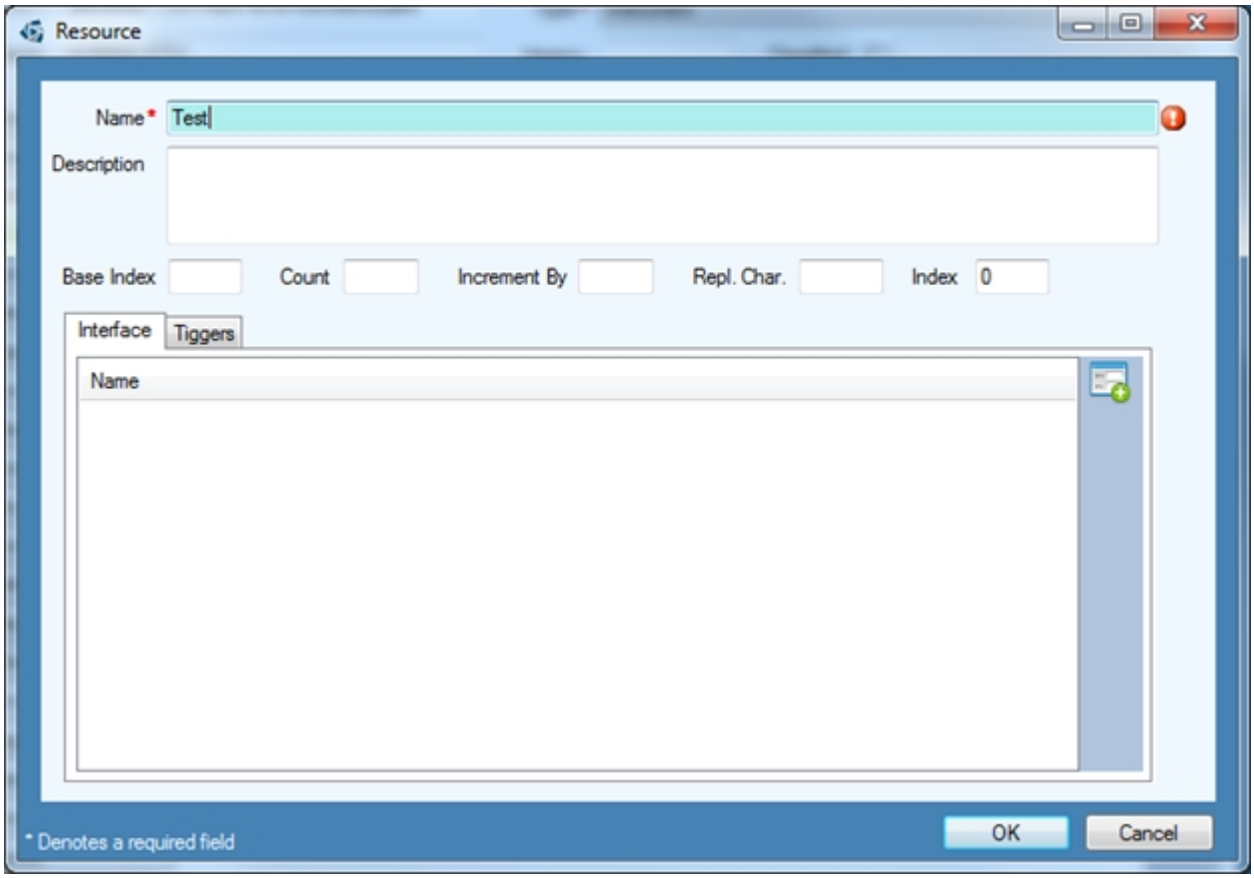
Power On Defaults

Shall document the item's power-on default state.

Resources

Resources

Shall document the logical entities within the item that provide source, sensor or load capabilities and map signals to ports.



Name **(Required)**
A descriptive or common name for the resource.

Description
A written representation that accurately depicts the resource.

Base Index
Staring index for the items.

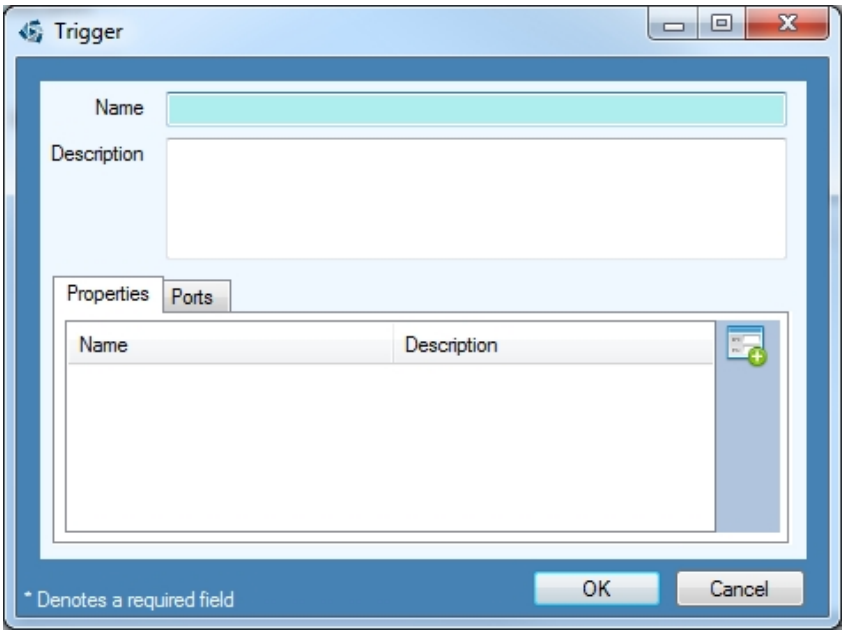
Count
Number of identical items.

Increment By
A positive integer that represents an incremental value on the index.

Repl. Char.
Specifies the character replacement in association with the calculated index.

Index
The value of the index.

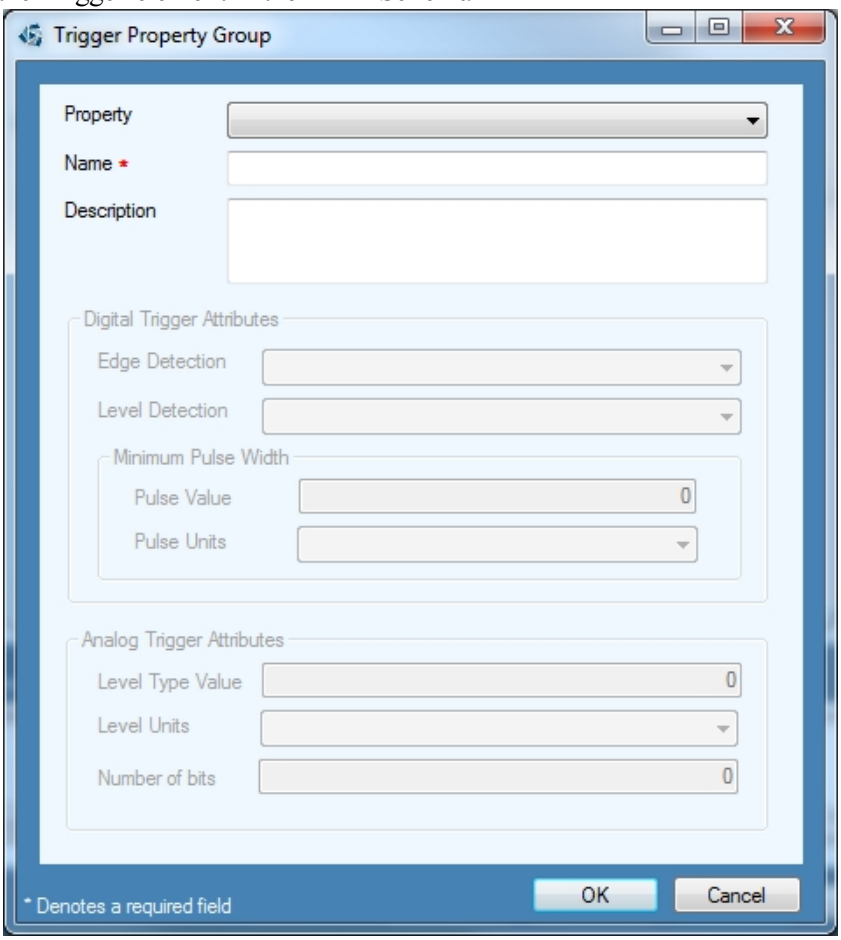
Triggers
A child of the Resources element in the XML Schema



Name
A descriptive or common name for the trigger.

Description
A written representation that accurately depicts the trigger.

Trigger Property Group (Properties)
This is a child of the Trigger element in the XML Schema



Property
Where the trigger property group shall be selected.

Name
A descriptive or common name for the trigger property.

Description
A written representation that accurately depicts the trigger property.

Requirements

Requirements

Shall be used to document the calibration, operational, environmental, and power requirements of the item.

Software

Software

Shall be used to document the software present on the item.

Specifications

Specifications

When present, shall document the specifications of an instrument.

Status Codes

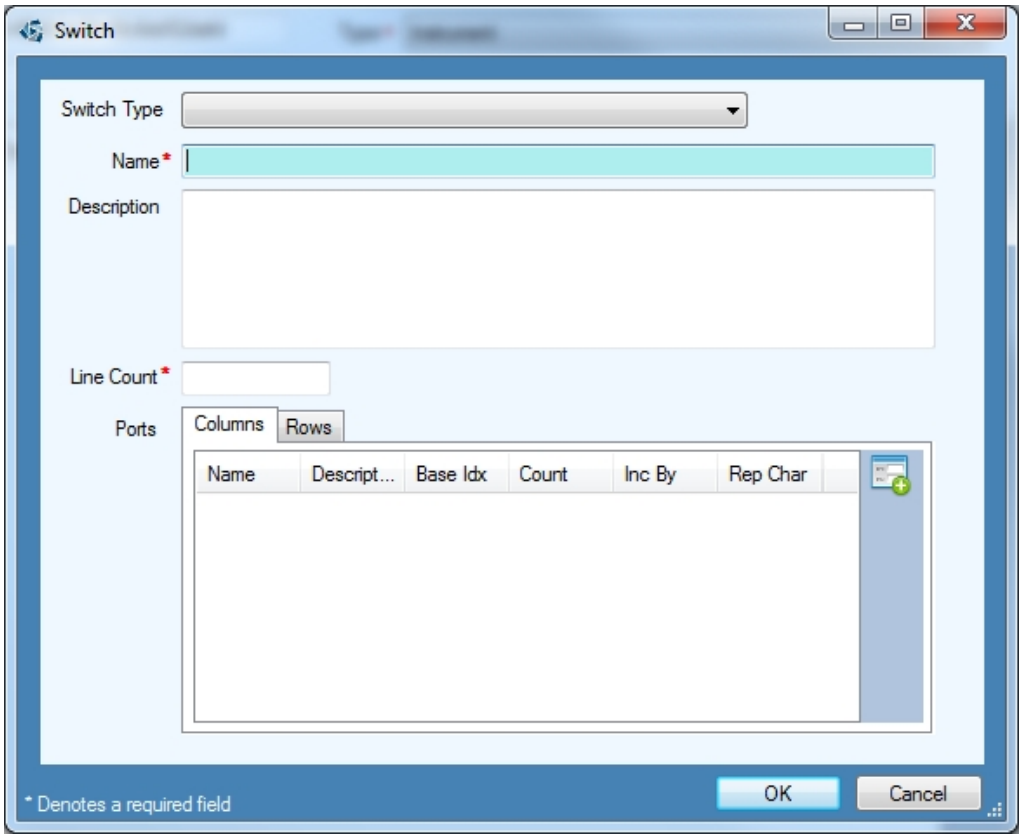
Status Codes

When present, shall contain an unbounded set of status code elements that present the status or built-in-test (BIT) code data associated with the UUT.

Switching

Switching

This complex type shall document the properties of a switching subsystem.



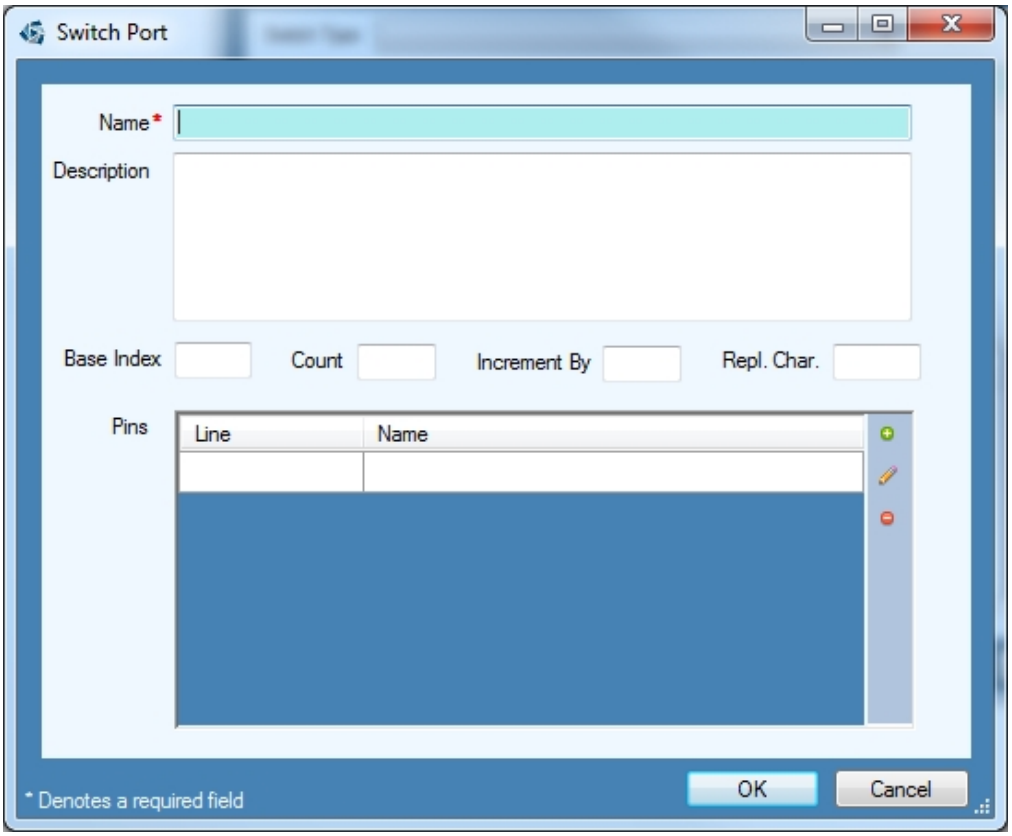
Switch Type
This is where the type of switch shall be selected.

Name (Required)
A descriptive or common name for the switch.

Description
A written representation that accurately depicts the switch.

Line Count (Required)
The number of matrix lines available to connect the rows and columns.

Switch Port (Columns, Rows**)**
A child from the Switching element in the XML Schema



Name **(Required)**
A descriptive or common name for the resource.

Description
A written representation that accurately depicts the resource.

Base Index
The starting index for the items

Count
The number of identical items.

Increment By
A positive integer that represents an incremental value on the index.

Repl. Char.
Specifies the character replacement in association with the calculated text.

Pins
This element shall contain the line and name of the pin.

Terminal Blocks

Terminal Blocks

Shall identify the terminal blocks within the test equipment.

Warnings

Warnings

When present, shall contain any hazard or warning information related to the operation of the UUT.

ATML

Automatic Test Markup Language (ATML) is an IEEE family of standards that creates a common format for the exchange of Automatic Test Equipment (ATE) and test information. ATML is based on eXtensible Markup Language (XML), which uses text files that are both human- and machine-readable to exchange structured data between applications. The ATML Standards include so-called "Dot" standards that describe each element of ATE.

- 1671.1 Test Description documents the test strategy, i.e., which signals the ATE need to send to the UUT and what responses are expected from the UUT.
- 1671.2 Instrument Description documents the instrument's physical characteristics, operating requirements, signal capabilities, and other pertinent information.
- 1671.3 Unit Under Test Description documents the UUT's physical characteristics, operating requirements, interface requirements, and other pertinent information.
- 1671.4 Test Configuration documents the instruments and adapters required to execute the test description.
- 1671.5 Test Adapter Description documents the interface device's physical characteristics, operating requirements, signal capabilities, and other pertinent information.
- 1671.6 Test Station Description documents the test station's physical characteristics, operating requirements, instruments, signal capabilities, and other pertinent information.

In addition, the ATML Standard references several other IEEE standards and schema. These include:

- 1641 Signal and Test Definition Standard
- 1636 Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA)
- Common.xsd
- HardwareCommon.xsd
- TestEquipment.xsd

These documents are available from the IEEE (www.ieee.org).