

Create XML Library

To run this script

Select **Scripts >Project > Create_XML_Library** from the Menus.

Or, in versions that support the script, you can run the utility directly from this page using this button.

Run Utility

Description

This script helps you create a library of parts for Microwave Office and Visual System Simulator. You configure the parts you would like them in an AWRDE project and the script will write out all the necessary information to an XML format that is used to describe parts libraries. It is much easier to configure the parts in a system diagram than editing XML files directly. You might use this script for various purposes including:

- Creating a common list of models you want to configure in the Element tree for easy access.
- Saving the state of a model that was difficult to configure for future use.
- Managing approved vendor models to create a library that could be shared by your companies designers.

To use this Script

1. Run the script as described above.
2. A new directory named "Library" is created in the same directory as your AWRDE project. This directory has all the files needed for your library.
3. A log is written named "log.txt" in the same directory as your project. It is also imported into the project when the utility is done. This log contains any errors, warnings and information messages including the name and location of files written. An example log is shown below.

AWR Utilities
Create XML Library--To run this script

Information, Errors, and Warnings Found:

Warning: process libraries found in this project which can have symbol files loaded. If you are generating this library for a PDK, this is fine. If you are making a generic library of parts, this could make symbols available to use that are not available to other people.

INFO: Creating top level circuit element xml file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Circuit_Elements.xml. From the xmluser (Help > Show Files/Directories) Directory, add an xml file in the Circuit Elements directory that points to this file to hook up this Library.

INFO: Creating file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Lumped_Elements.xml

Warning: Found both an element partnumber and an user attribute named partnumber, element part number used: IND.L1 found in document: Lumped Elements

Error: Found unknown distribution type : CAP.C1 found in document: Lumped Elements

Error: If using statistics, must specify a statistical variation value: CAP.C1

Error: Libraries cannot use custom symbols in the project, found on : CAP.C2 found in document: Lumped Elements

Error: Found model parameter other than a number, could be variable or different parameter type: RES.R2 found in document: Lumped Elements

Warning: Found library using dxf on cell for : CAP.dxf found in document: Lumped Elements. This can cause problems when importing into projects with different units potentially.

Warning: ports not written out, found PORT.P1 found in document: Lumped Elements

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Murata.xml

Error: No description found for a subcircuit, suggest using a user attribute to give a discription of the model, found for: SUBCKT.S1 found in document: Murata

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Sources.xml

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\fet.xml

INFO: Creating file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\subcircuits.xml

Error: NPORT elements not supported: NPORT_F.X1 found in document: subcircuits

Error: No description found for a subcircuit, suggest using a user attribute to give a discription of the model, found for: SUBCKT.S1 found in document: subcircuits

INFO: Creating file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\substrate.xml

Error: Code currently doesn't support STACKUP elements, did not write: STACKUP.SUB2 found in document: substrate

INFO: Creating top level system element xml file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\System_Elements.xml. From the xmluser (Help > Show Files/Directories) directory, add an xml file In the System Blocks directory that points to this file To hook up this Library.

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\ADI.xml

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Filters.xml

INFO: Creating file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\MiniCircuit_LNA.xml

INFO: Creating file: C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Mixers.xml

INFO: Creating file:

C:\svn\xmllibs\trunk\Scripts\xml_generate_scripts\Library\Oscillators.xml

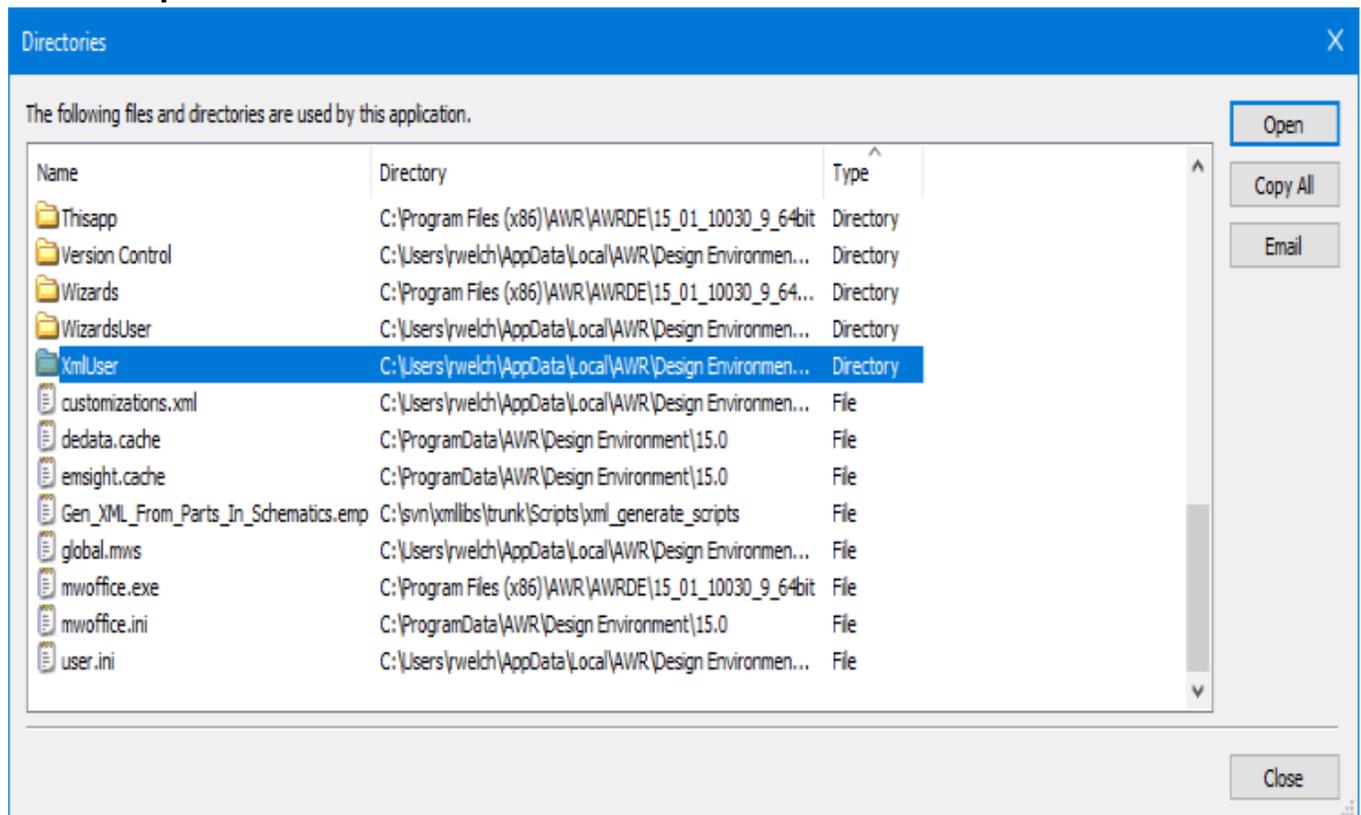
Total Errors found: 8

To use this library in the AWR Design Environment

Once your library is complete, you need to hook up your library for the AWR Design Environment to find it.

To do this:

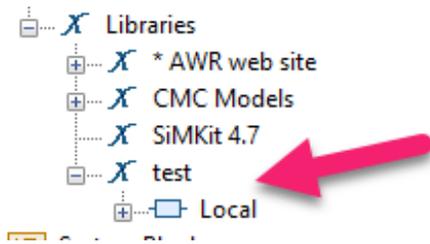
1. Select **Help > Show Files/Directories** and then double click on the **XmlUser** folder.



2. For Microwave Office libraries, open the **Circuit Elements** folder and add a new xml file. The file contents should be similar to below.

```
<?xml version="1.0"?>
<XML_COMPONENT_DATA xmlns="urn:awr-lib-data">
<FILE Name="Local" Icon="FOLDER"
>C:\xmllibs\trunk\Scripts\xml_generate_scripts\Library\circuit_elements.xml</FILE>
</XML_COMPONENT_DATA>
```

If this file was named "test.xml" in this folder, then the library will display in the software as shown below.



Where the xml file name will the name of the top node as shown above and then second node is the "Name =" in the XML file.

3. For VSS libraries, open the **System Blocks** folder and add a new xml file and follow the same steps for the Circuit Elements above.

Script Details

Folder Structure

The name of the schematic or system diagram will specify the name of the folder for those parts in the element tree.

In this example, the system diagrams look like the following:

The screenshot shows four panels from the AWR Utilities interface, each displaying configuration parameters and circuit diagrams for different components:

- MiniCircuit LNA:**
 - MiniCircuit LNA list: http://www.minicircuits.com/products/amplifiers_smt_low_noise.shtml
 - AMP_F ID=SAV_541p_VDS3V DATA=SAV_541p_VDS3V NOISE=Auto RFIFRQ=
 - AMP_F ID=SAV_541p_VDS4V DATA=SAV_541p_VDS4V NOISE=Auto RFIFRQ=
- Oscillators:**
 - ADR4350 (Ab) : ports http://www.analog.com/static/imported-files/data_sheets/ADR4350.pdf
 - ID=ADR4350_2200MHz
 - FRQ=2.2 GHz
 - PWR=-7 dBm
 - PHS=0 Deg
 - CTRFREQ=
 - SMRFRQ=
 - ZS=_Z0 Ohm
 - TN=_TAMB DegK
 - NOISE=Auto
 - PNMASK="ADR4350_PN_2200MHz"
 - PNOISE=Generate phase noise.
- Mixers:**
 - ID=ADL5385 MODE=SUM LOMULT=1 FOOUT= RFIFRQ= GCONV=-33 dB PTOB=3 dBm IP3=21 dBm LO2OUT=-21 dB IN2OUT=-36 dB LO2IN=-25 dB OUT2IN=-25 dB PLO=-7 dBm PLOUSE=Spur reference only PIN= PINUSE=IN2OUTH Only NF=14 dB NOISE=Auto
 - ID=ADL5387 MODE=DIFF LOMULT=1 FOOUT= RFIFRQ= GCONV=-3.3 dB PTOB=12.3 dBm IP3=27.4 dBm LO2OUT=-125 dB IN2OUT=-20 dB LO2IN=-75 dB OUT2IN=-70 dB PLO= PLOUSE=Spur reference only PIN= PINUSE=IN2OUTH Only NF=16.5 dB NOISE=Auto
- Filters:**
 - LIN_S ID=Filter_S2P1 NET="FHX35LG" INPORT={1} OUTPORT={2} NOISE=Auto

After the script is finished and the library configured, notice how the System Diagram names are the names of the nodes in the library.

The screenshot shows a hierarchical tree of libraries. The 'My Library' folder is expanded, showing sub-folders: Filters, MiniCircuit LNA (highlighted), Mixers, and Oscillators. Below the tree is a table with two columns: 'Models' and 'Description'.

Models	Description
-SAV_541p_VDS3V	Frequency Dependent Behavioral Amplifier,
-SAV_541p_VDS4V	Frequency Dependent Behavioral Amplifier,

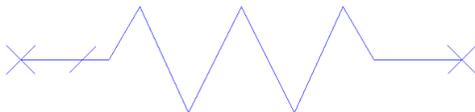
Name

The name is value shown up in the **Models** column shown below.

Models	Description
~ R1	Resistor (Closed Form)
~ AwrlInductor	Custom Inductor
~ C1	Capacitor (Closed Form)
~ C2	Capacitor (Closed Form)
~ R2	Resistor (Closed Form)

This value can come from either the model ID parameter or a user attribute. For example in the list above, the first model came from the ID parameter from the resistor below.

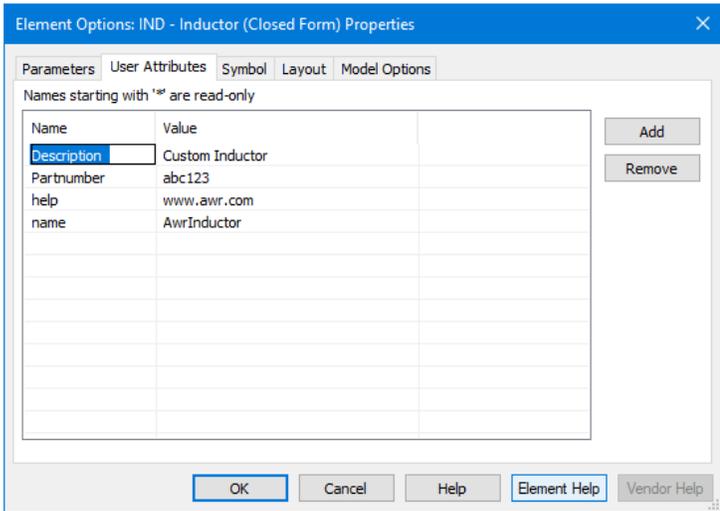
```
RES
ID=R1
R=1 Ohm
```



Alternately, you can define a user attribute with **Name=name**. If this value is set, it will always be used instead of the ID.

To access the user attributes, double click on the element and select the "User Attributes" tab, as

shown below.

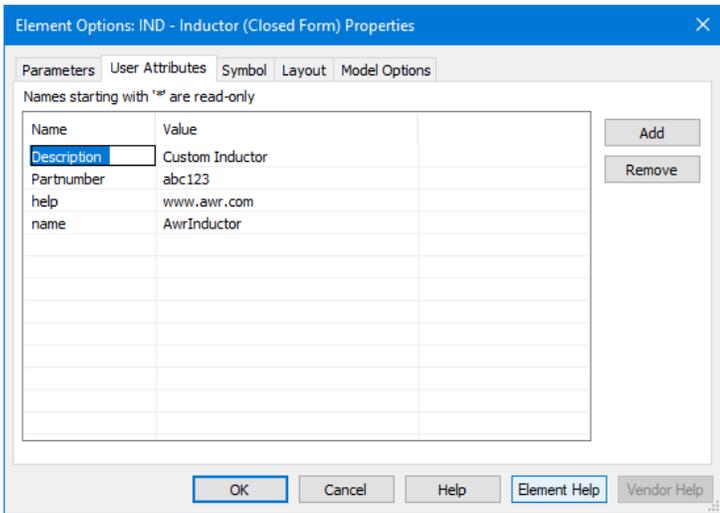


Description

The description is value shown up in the **Description** column shown below.

Models	Description
~ R1	Resistor (Closed Form)
~ AwrInductor	Custom Inductor
~ C1	Capacitor (Closed Form)
~ C2	Capacitor (Closed Form)
~ R2	Resistor (Closed Form)

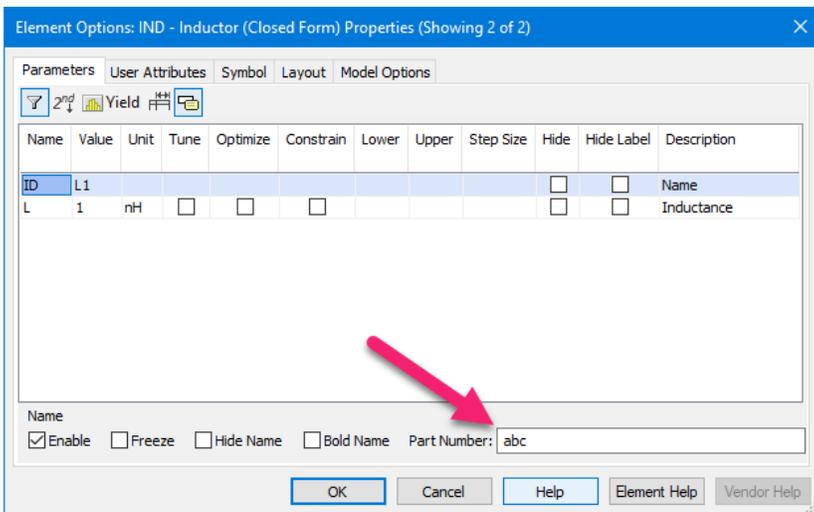
This value can come from a user attribute with **Name=Description**. For example in the list above, the second model is defining this description with the attribute shown below.



If this value is not set, the value will be the description of the model used for the part in the AWR Design Environment. One special case is a subcircuit. Since it doesn't have model description, the log will report that a description is missing.

Part Number

The part number is a field located on each element as shown below.



This value can come from either the **Part Number** field shown above or the user attribute with **Name=Partnumber**. If both are set, an error is issued and no value is set.

Help

By specifying a help url, when the user double clicks on the element, the **Element Options** will have the **Vendor Help** button enabled that will take you to that help as well as when you right click on a model in the Element tree and select **Element Help...**

This value comes the user attribute with **Name=help** If both are set, an error is issued and no value is set.

Symbols

You can assign different symbols to your library parts, this is especially useful when using a subcircuit. You don't need to do anything else for this to get captured in the library. The only limitation is that the symbol must be from the symbols shipped with the software. If you use a symbol created in the project, the utility will issue an error in the log.

Cells

You can assign layout cells to your library parts. You don't need to do anything else for this cell to get captured in the library. The only limitation is that the cell must be from a GDSII library. DXF libraries don't fully understand the layout units and can cause serious issues if the user of the library doesn't select the right units.

User Attributes

As shown, user attributes are used to specific specific parts of the library. You may specify any other user attribute that makes sense for your library of parts and those attributes will be retained in the library..