Scripting How-To: Converting Between MKS and User Units in AWR Scripting

Summary

If your API application displays any sort of User Interface (UI) you may find that you need display numeric values to the user. By default the API works directly in base MKS units but the user may be more familiar with some project specific units. For example when working on chip design the user may want to work with lengths in terms of microns (1e-6 meters) and if they are working on board designs they may prefer mils (1/1000 inch). Since the API always deals in base units you may want to convert numeric values to user units before displaying them. And when getting values from the user you may want to convert them back from user units to base units.

In order to facilitate these types of conversions the API supports a Project.Units collection which provides multiplication factors for converting between base units and user units. In addition items in the collection also contain unit strings which provide a text representation of the unit factor to be placed after the numeric value. For example, lets say we have a project that has length units in Microns and we want to present 5e-6 meters in user units. Lets use the immediate window in the scripting environment and try a conversion.

Code Snippets

```
? 5e-6 / Project.Units(mwUT_Length).MultValue & Project.Units(mwUT_Length).UnitString
"5um"
```

So that gave us 5um or 5 microns. Lets look at the parts. First we used the used the units collection off the project Project.Units. This from this collection we used an index defined by mwUT_Length to get the unit entry for length. And then we asked for the MultValue of that unit. The MultValue is the value needed to go from user units to base units so:

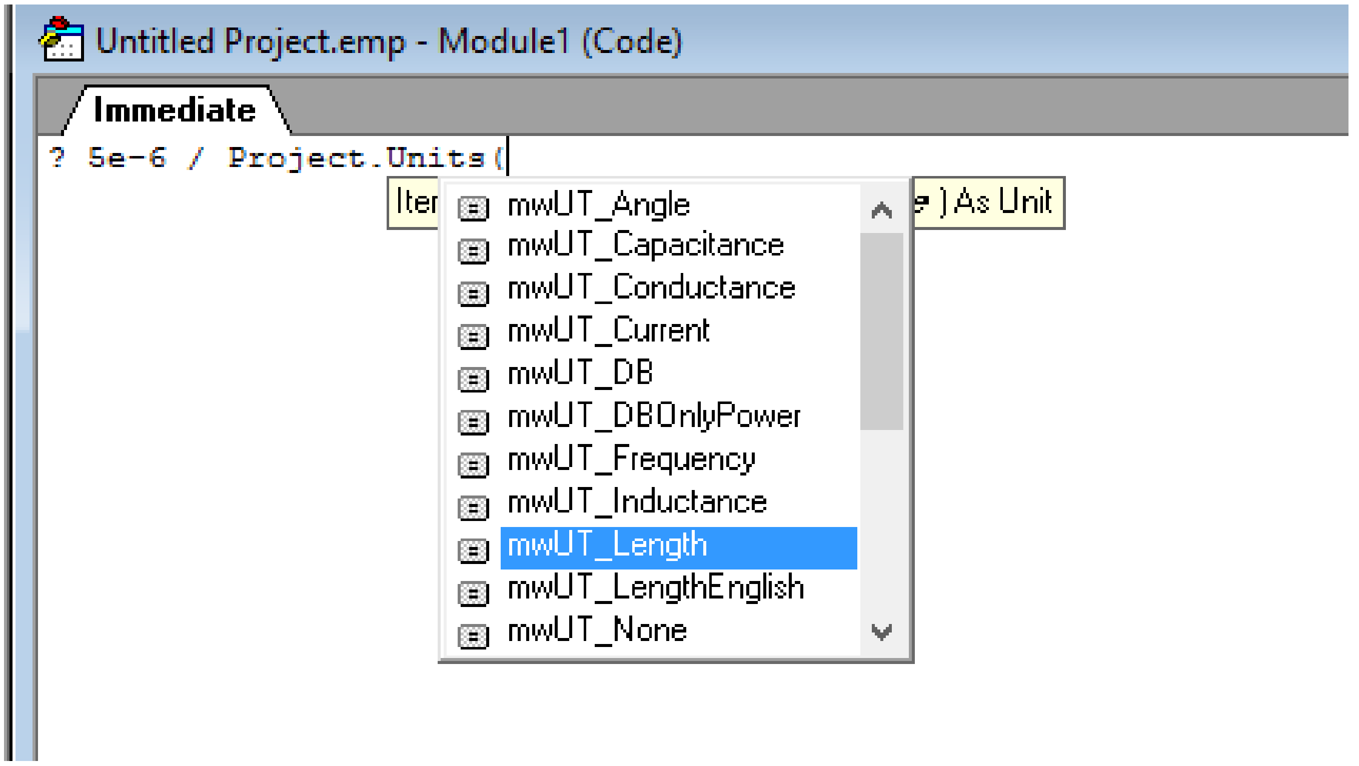
```
Base Units =  User Units * Mult Value
```

Or in the case of the example above:

```
User Units = Base Units / Mult Value
```

Next we added the UnitString on to the end so the user would know that this value is being presented in microns which gives us the "um" part of the output.

One of the other neat features about the 'Units' collection is that the index variable of the Units.Item() method is a mwUnitType so intellisense in the scripting environment will help you pick the correct value:



```
? 5e-6 / Project.Units(
```

In addition all the unit types available this table:

Unit Type	Type Index
mwUT_None	0

mwUT_Frequency	1
mwUT_Capacitance	2
mwUT_Inductance	3
mwUT_Resistance	4
mwUT_Conductance	5
mwUT_Length	6
mwUT_LengthEnglish	7
mwUT_Temperature	8
mwUT_Angle	9
mwUT_Time	10
mwUT_Voltage	11
mwUT_Current	12
mwUT_PowerLog	13
mwUT_Power	14
mwUT_DB	15
mwUT_String	16
mwUT_Scaler	17
mwUT_DBOnlyPower	18
mwUT_WattsOnlyPower	19