

Python Scripting How-To: How to Parse Measurement Strings

When you get a measurement object there is a lot of information packed into it including the data source, the name of the measurement, the plot dimensions, the type of the axes, etc. Here is a utility class that gets and decodes all this information from a measurement.

```
class AwrMeasurement():
    def __init__(self, measurement: mwo.CMeasurement):
        (source, name, simulator) = self.parse_meas_string(measurement.Name)

        self.name = name
        self.source = source # schematic name for example
        self.simulator = simulator
        self.data_type = mwo.mwMeasDataType(measurement.DataType)._name_
        self.plot_dim = measurement.PlotDimension
        self.x_units = mwo.mwUnitType(measurement.UnitType(1))._name_
        self.y_units = mwo.mwUnitType(measurement.UnitType(2))._name_

    @classmethod
    def parse_meas_string(cls, s: str) -> Tuple[str, str, str]:
        (source, name) = s.split(':')
        if '.' in source:
            (source, simulator) = source.split('.')
        else:
            simulator = ''
        return (source, name, simulator)
```

Using this you can parse a measurement string using:

```
(source, meas_name, simulator) = AwrMeasurement.parse_meas_string('Pulse Signal.HS:Vtime(PORT_1,1)')
```

Or create an AwrMeasurement object by passing in a CMeasurement object as in:

```
m = awrde.Project.Graphs[0].Measurements[0]
parsed_measurement = AwrMeasurement(m)
```