THE QUICK START GUIDE FOR ARRAY HARDWARE

Hello, I am Mike Walker.
Welcome to a demostration of how to find hardware information in an SMI-S 1.8.0 Array. This demo is shown as a Quick Start Guide for Array hardware on my GitHub site for SMI-S Mocks, identified in the video on installation and setup for running mock ups of SMI-S WBEM Servers.
The quick start guide 7-page document highlights information that can be found in about 30 pages of SMI-S, the Physical Package in the Common Book and disk drive lite profile in the block book of SMI-S.
In this script we will be working with pywbem 1.1.1 and version 0.8.0 of pywbemtools

(pywbemcli).
So, let's begin. We will switch to my command prompt window.

First, we go to our virtual environment for mocks:

```
C:\Users\FarmerMike> workon mocks
```

We are now in our virtual environment for mocks.
We will be working with a mock server that supports an SMI-S 1.8.0 Array. So, we need to establish a connection to the
Array mock up:

```
(mocks) c:\Users\FarmerMike\devenv>pywbemcli -o table -n ArrayMock
```

In our command, I requested the default format of output to be in “table” format (-o table) and name the mock that
I want (-n ArrayMock). The command worked and we get a pywbemcli prompt to start entering commands on the ArrayMock.

PHYSICAL PACKAGES

The elements that reports hardware information are instances of CIM_PhysicalPackage (and its subclasses).
So, we start by looking for classes that report hardware information (CIM_PhysicalPackage)

```
pywbemcli> instance enumerate CIM_PhysicalPackage --pl Tag,Manufacturer,Model
```

There are 10 hardware packages in this Array. We can see the manufacturer and model, but it does not tell us what the
hardware does or what it supports.

WHAT THE HARDWARE SUPPORTS

The next question to answer is what part of the Array does the hardware support.
We will determine this by iterating on the following command to determine logical elements supported by the hardware.

```
pywbemcli> 0
```

We will start with the first one (selection 0):

```
Input integer between 0 and 9 or Ctrl-C to exit selection: 0
```

The first two associations will be talked about later. The third association (CIM_SystemPackaging) links the physical package to the Array System. Also notice that the CreationClassName is a CIM_Chassis.
CIM_Chassis is a subclass of CIM_PhysicalPackage.
Now let’s run the same command and select the second item in the list:

```
pywbemcli> -o mof instance shrub CIM_PhysicalPackage.?
```

And we select item 1:

```
Input integer between 0 and 9 or Ctrl-C to exit selection: 1
```

In this case the physical package shrub ONLY includes the CIM_Container association. The thing it represents is a rack (the GroupComponent in the first Container association) for a set of packages (the PartComponents) and the rack itself is contained in the Chassis for the Array.

Now we will at the shrub for the third physical package.

```
pywbemcli> -o mof instance shrub CIM_PhysicalPackage.?
```

So, we will select item 2:

```
Input integer between 0 and 9 or Ctrl-C to exit selection: 2
```

The first association (CIM_Container) says the package is contained in the rack. The second association (CIM_Realizes) links the physical package to a Disk Drive. So, the package is a Disk Drive. And it appears the same is true for the next 7 packages.