THE QUICK START GUIDE FOR ARRAY SOFTWARE
Hello, I am Mike Walker.
Welcome to a demostration of how to find software information in an SMI-S 1.8.0 Array. This demo is shown as a Quick Start Guide for Array software 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 is a 6-page document that highlights information that can be found in about 32 pages of SMI-S, the software and software inventory profiles in the Common book and the disk drive lite profile in the block book of SMI-S.
In this demo 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.

SOFTWARE IDENTITIES
The elements that reports software information are instances of CIM_SoftwareIdentity (and its subclasses).
So, we start by looking for classes that report software information (CIM_SoftwareIdentity)

```
pywbemcli> instance enumerate CIM_SoftwareIdentity --pl
InstanceID,VersionString,Manufacturer,MajorVersion,MinorVersion,RevisionNumber
```

We see we have 10 software elements from 3 software manufacturers.

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

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

We will start with the first software element by selecting item 0

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

We see the ACME Software has a CIM_ElementSoftwareIdentity to a set of profiles. These are the Array profile and
its component profiles. The ACME software is the provider code for the Array. The ACME Software also has a
CIM_InstalleddSoftwareIdentity to a Computer System. The system is the top level system of the Array. This means the
software is installed on the array.
Let’s move on to the second software element:

pywbemcli> -o mof instance shrub CIM_SoftwareIdentity.?

We will now get information for the second software element by selecting item 1

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

We see the ByteStor Software has a CIM_ElementSoftwareIdentity to a disk drive. The ByteStor software is the firmware for
the disk drive.
Notice that the ByteStor Software does NOT have a CIM_InstalleddSoftwareIdentity to anything. This is because it would be
redundant with the CIM_ElementSoftwareIdentity. That is, the firmware is on the disk drive.

Let’s move on to the last software element:

We will now get information for the last software element by selecting item 9

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

We see the Management Server Software has a CIM_ElementSoftwareIdentity to a set of profiles. These are the WBEM Server
and SNIA Server profiles and their component profiles. The Management Server software is the provider code for the
interop namespace. The Management Server Software also has a CIM_InstalleddSoftwareIdentity to a System. The system is
the system of the server profiles. This means the software is installed on the server system.