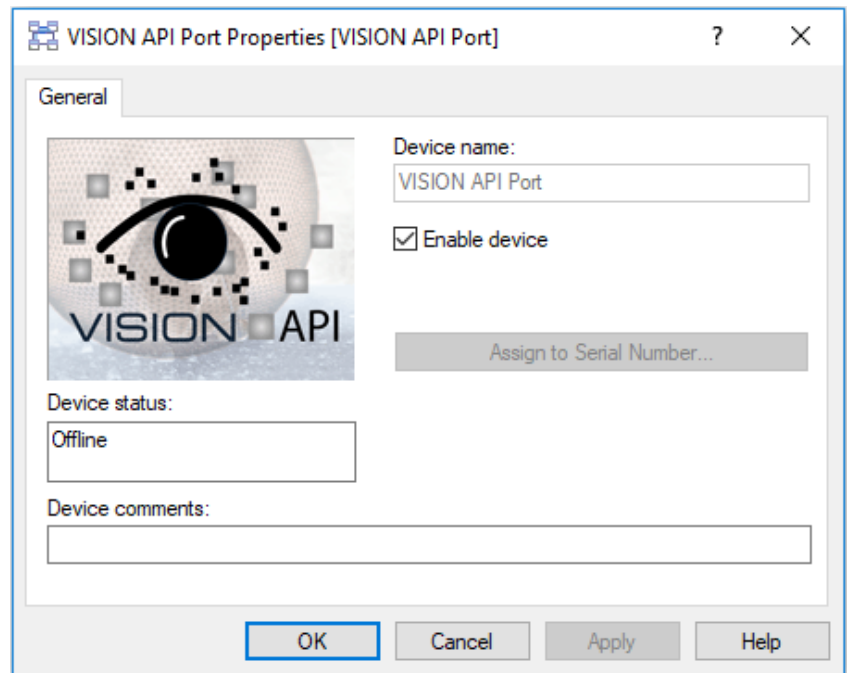


Virtual PCM and API Scripting

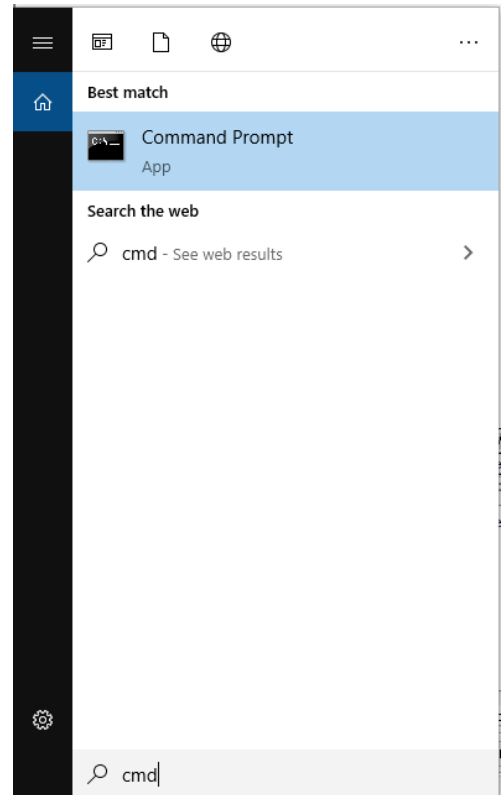
The Virtual PCM is a virtual power control module for use with a VISION API Port. For example, data can be written to Virtual PCM items from an external application, for the purpose of monitoring and recording in VISION. The Virtual PCM mimics a real ECU interface therefore, VISION can record the data item values directly into a recorder file to correlate the script results with data obtained from the real ECU.

The Virtual PCM device can be directly manipulated through the VISION API via a user developed script. This script can update Data Items defined in the Virtual PCM as required for the user's purpose. This may include real-time signal processing tasks such as Integration, signal down sampling, or logging of events and conditions as determined by the user script.

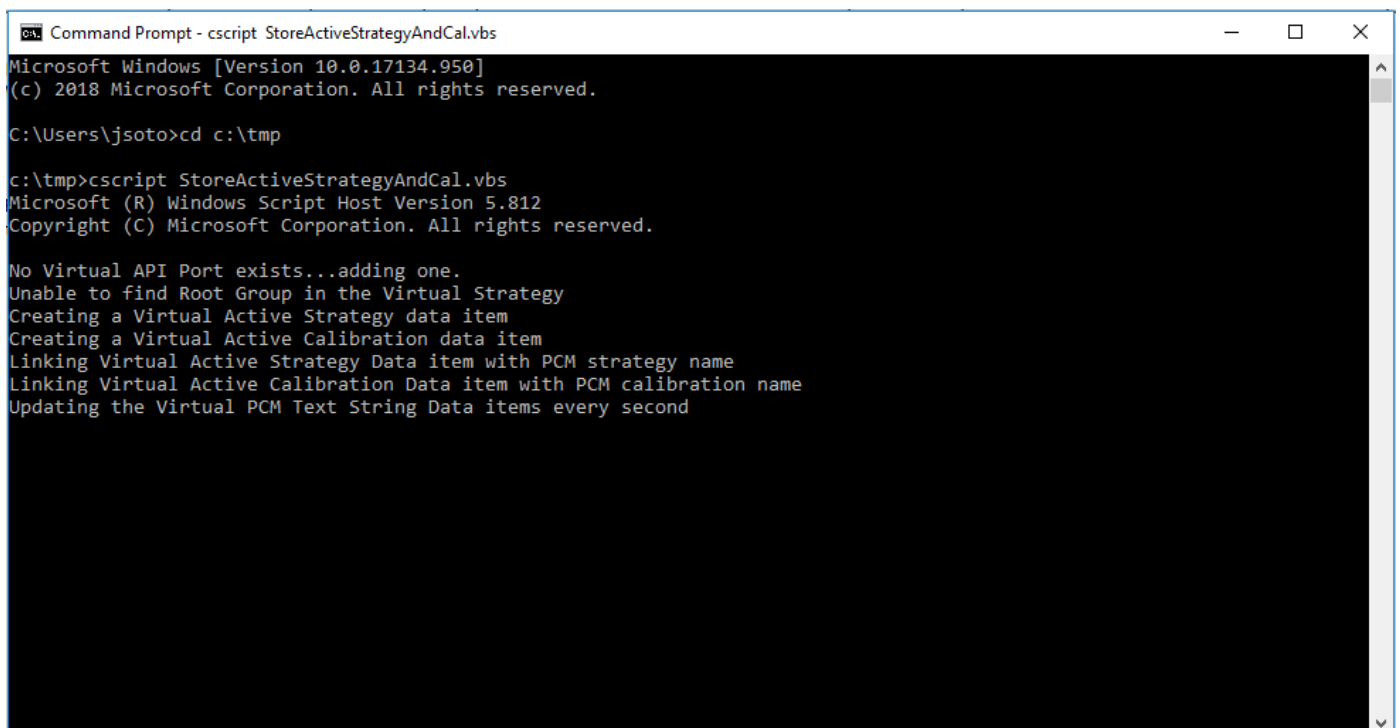
Correlating the script results with data obtained from the real ECU enables a use beyond writable items. For example, having calibration items used to control the execution of a script, or hosting a 2D/3D table that does not exist in an ECU but takes inputs from another device to result in an output. The Virtual PCMs capabilities are limited only by the availability of PC resources.



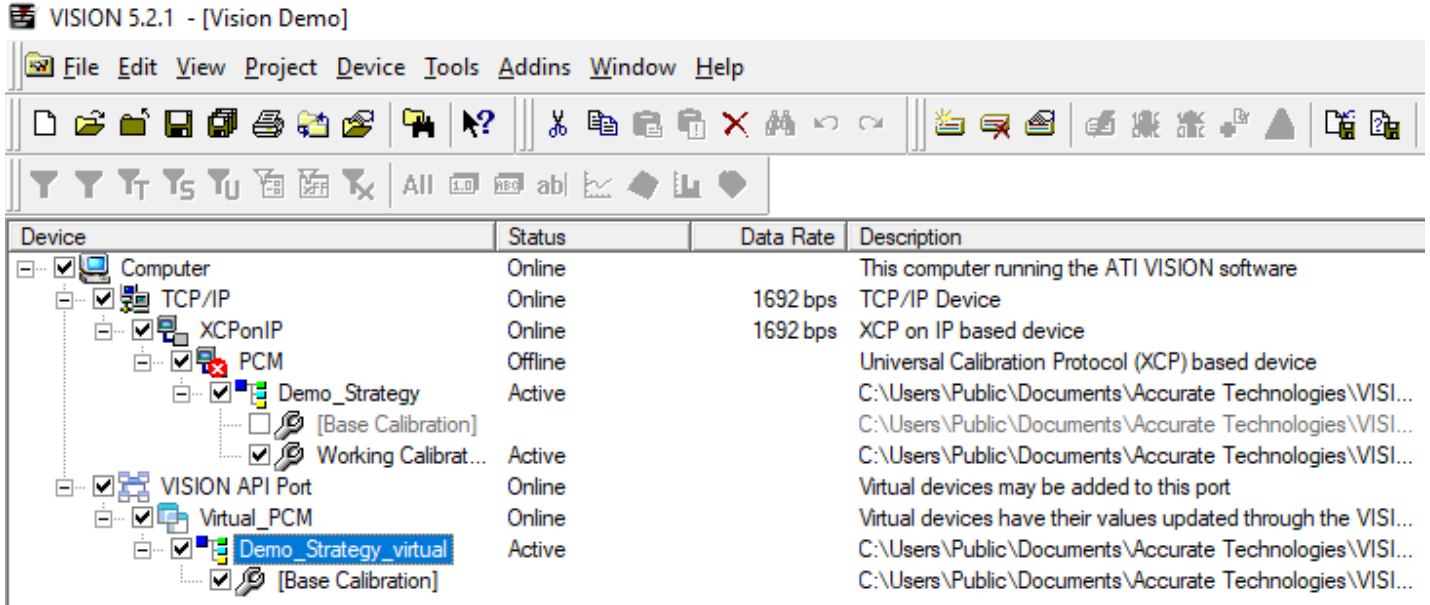
Although many scripting languages can be used to access the VISION API, the sample script provided with this tech tip is written in the vbscript language. And to make use of the example to create a Virtual PCM, open a Command Prompt from the start menu in Windows. This can also be found by typing 'CMD' into the Windows Search menu.



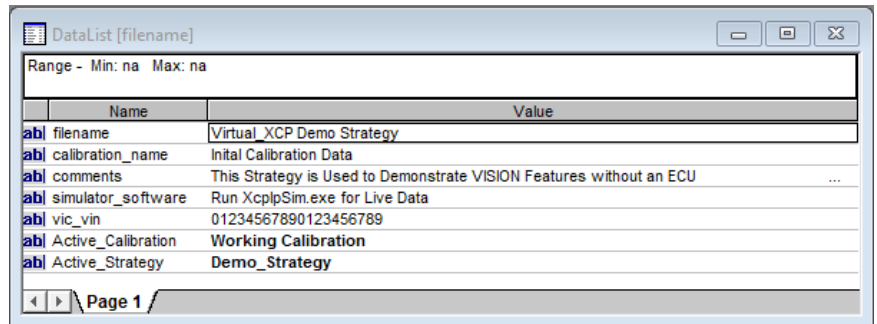
To apply a user developed script to the Virtual PCM start by opening the Command Prompt menu from windows. This can be found by typing "CMD" into the windows search menu.



The VISION project will then be automatically populated with a VISION API port and Virtual PCM if these devices don't already exist in the VISION project.



The two data items (of Text String type) are created and will be periodically updated with the values of the Real PCM's "Active Calibration" and "Active Strategy" values. As long as the VISION project is online these Text String scalars in the "Virtual PCM" will be updated with the names of what calibration and strategy is active in the PCM device.



Below is an example of a user developed script for the Virtual PCM, the example is of the type vbscript:

```
Const VISION_DATAITEM_STRING = 12
Const VISION_DEVICE_VIRTUALPCM = 36
Const VISION_DEVICE_APIPORT = 37

Set objStdOut = WScript.StdOut

Dim Project
Set Project = CreateObject("Vision.ProjectInterface")

Dim V_App
Set V_App = CreateObject("Vision.Application")

Dim StrategyFileIntf
Set StrategyFileIntf = CreateObject("Vision.StrategyFileInterface")

Dim Group_Interface
Set Group_Interface = CreateObject("Vision.GroupInterface")

Dim Group_Intf2
'Set Group_Intf2 = CreateObject("Vision.GroupDataItemInterface")

if Project.IsOpen then

    Dim Device
    Set Device = Project.FindDevice("PCM")

    Dim Virtual_API_Port
    Set Virtual_API_Port = Project.FindDevice("VISION API Port")

    Dim Virtual_PCM_ActiveStrategy
    Dim Virtual_PCM_ActiveCalibration

    ' If not Virtual API Port exists, add everything
    if Virtual_API_Port is Nothing then
        objStdOut.WriteLine "No Virtual API Port exists...adding one."

        'Create a new Virtual PCM
        Dim CurrentProject
        Set CurrentProject = V_App.GetCurrentProjectInterface
```

```

Dim MyRootDevice
Set MyRootDevice = CurrentProject.RootDevice

Set Virtual_API_Port = MyRootDevice.AddDevice(VISION_DEVICE_APIPORT)

if Virtual_API_Port is Nothing then
    objStdOut.WriteLine "Unable to add a Virtual API Port - exiting!!"
else
    Dim Virtual_Device
    Set Virtual_Device = Virtual_API_Port.AddDevice(VISION_DEVICE_VIRTUALPCM)

    If Virtual_Device is Nothing then
        ObjStdOut.WriteLine "Unable to add a Virtual PCM - exiting!!!"
    else
        Dim ActiveStrategy
        Set ActiveStrategy = Device.ActiveStrategy

        Dim Virtual_ActiveStrategy
        Set Virtual_ActiveStrategy = Virtual_Device.ActiveStrategy

        if Virtual_ActiveStrategy is Nothing then

            FullStrategyFileName = Project.Directory + "\" + Device.Name + "\" + ActiveStrategy.Name
+ ".vst"
            Virtual_Device.AddStrategy(FullStrategyFileName)

            BaseVirtualFile = ActiveStrategy.Name + "_virtual"
            VirtualFileName = BaseVirtualFile + ".vst"

            FullVirtualStrategyFileName = Project.Directory + "\" + Virtual_Device.Name + "\" +
VirtualFileName
            Virtual_Device.ActiveStrategySaveAs(FullVirtualStrategyFileName)

            Virtual_Device.RemoveStrategy(FullVirtualStrategyFileName) 'Close and reopen to refresh
the strategy (and name)
            Virtual_Device.AddStrategy(FullVirtualStrategyFileName) 'to make VISION happy and
stable.
        end if

        FullVirtualStrategyFileName = Project.Directory + "\" + Virtual_Device.Name + "\" +
VirtualFileName
        StrategyFileIntf.Open(FullVirtualStrategyFileName)

```

'Open the Virtual Strategy and Delete all of the unnecessary data item

```
Dim BaseGroup
```

```
Set BaseGroup = StrategyFileIntf.GroupDataItem
```

```
if not BaseGroup is Nothing then
```

```
    objStdOut.WriteLine "Unable to find Root Group in the Virtual Strategy"
```

```
    if BaseGroup.FindDataItem("Active_Strategy") is Nothing then
```

```
        objStdOut.WriteLine "Creating a Virtual Active Strategy data item"
```

```
        BaseGroup.CreateItem VISION_DATAITEM_STRING, "Active_Strategy"
```

```
    else
```

```
        objStdOut.WriteLine "Virtual Active Strategy data item already exists!!"
```

```
    end if
```

```
    if BaseGroup.FindDataItem("Active_Calibration") is Nothing then
```

```
        objStdOut.WriteLine "Creating a Virtual Active Calibration data item"
```

```
        BaseGroup.CreateItem VISION_DATAITEM_STRING, "Active_Calibration"
```

```
    else
```

```
        objStdOut.WriteLine "Virtual Active Calibration data item already exists!!!"
```

```
    end if
```

```
    StrategyFileIntf.Save
```

```
    Virtual_Device.ActiveStrategy.Reload
```

```
else
```

```
    objStdOut.WriteLine "Measurement Group Not found"
```

```
end if
```

```
Set Virtual_PCM_ActiveStrategy = Virtual_Device.FindTextString("Active_Strategy")
```

```
objStdOut.WriteLine "Linking Virtual Active Strategy Data item with PCM strategy name"
```

```
Virtual_PCM_ActiveStrategy.ActualValue = ActiveStrategy.Name
```

```
Set Virtual_PCM_ActiveCalibration = Virtual_Device.FindTextString("Active_Calibration")
```

```
objStdOut.WriteLine "Linking Virtual Active Calibration Data item with PCM calibration name"
```

```
Virtual_PCM_ActiveCalibration.ActualValue = ActiveStrategy.ActiveCalibration.Name
```

```
End If
```

```
End If
```

```
Project.Online = True
```

```
objStdOut.WriteLine "Updating the Virtual PCM Text String Data items every second"
```

```
do
```

```
    Wscript.sleep(1000)
```

```
    Virtual_PCM_ActiveCalibration.ActualValue = ActiveStrategy.ActiveCalibration.Name
```

```
    Virtual_PCM_ActiveStrategy.ActualValue = ActiveStrategy.Name
```

```
loop while Project.Online = True
```

```
objStdOut.WriteLine "Done looping for now!!"
```

```
end if
```

```
else
```

```
    objStdOut.WriteLine "No project open - please remedy"
```

```
End if
```

For additional questions, contact ATI support at support@accuratetechnologies.com.

Contact ATI Sales at : sales@accuratetechnologies.com

US +00 (1) 248 848 9200 / **China** +86 138 1023 6357 / **France** +33 (0) 1 72 76 26 10 / **Germany** +49 (0) 89 9700 7121

India +91 80 41 69 42 18 / **Japan** +81 3 6276 8950 / **Sweden** +46 (0) 31 773 7140 / **UK** +44 (0) 1767 652 340