Notes, cautions, and warnings

**NOTE:** A NOTE indicates important information that helps you make better use of your product.

**CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

**WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.
1 Introduction.......................................................................................................................... 5

2 Dell Precision Optimizer components.................................................................................. 6
   Uninstall Dell Precision Optimizer....................................................................................... 6

3 Performance........................................................................................................................... 8
   Policy Processing Engine....................................................................................................... 8
   Profile update tool................................................................................................................. 8
   Update options tool.............................................................................................................. 8

4 Track and Analyze.................................................................................................................. 9
   System analysis reports......................................................................................................... 9
      Report settings.................................................................................................................. 9
   Workload analysis............................................................................................................... 10
   CPU intelligence reports..................................................................................................... 10
   GPU intelligence reports..................................................................................................... 11
   System Diagnostics Report.................................................................................................. 11
   Performance Notifications.................................................................................................. 11
   Upgrade options.................................................................................................................. 12

5 System maintenance................................................................................................................ 13

6 User feedback........................................................................................................................ 14

7 Improve Dell Precision Optimizer........................................................................................ 15

8 Enterprise tools....................................................................................................................... 16
   WMI Providers..................................................................................................................... 16
   DPOCMD.EXE...................................................................................................................... 16
   Setup command line switches............................................................................................. 18
   SCCM.................................................................................................................................... 19
      Instructions for creating the Dell Precision Optimizer application package...................... 19
      Instructions for deploying application............................................................................. 19
      Verify deployment success in client systems.................................................................. 20
      Changing the Dell Precision Optimizer Client behavior using DPOCMD.EXE.................. 20
   SSRS Reports....................................................................................................................... 21
   KACE................................................................................................................................. 23
      Instructions for deploying Dell Precision Optimizer using KACE...................................... 24
      Changing Dell Precision Optimizer Client Behavior using DPOCMD.EXE.......................... 25
      Custom reports................................................................................................................. 26

A APPENDIX A - dpoCmd.exe Exit Codes.................................................................................. 28
Introduction

This document describes the tools, tips, and recommendations for the IT administrators to manage Dell Precision Optimizer remotely.
Dell Precision Optimizer components

The four main components of Dell Precision Optimizer are:

- Performance
- Track & Analyze Engine (TA)
- System Maintenance (SM)
- Dell Precision Optimizer Manager CLI (dpoCmd.exe)

Each of these components are implemented as a Windows service that also acts as a COM server. The Dell Precision Optimizer installer package installs the services along with Dell Precision Optimizer support DLL(s), user interface (UI) components, kernel mode device driver(s), and so on into the POA Installation folder. Additionally, a taskbar application may be installed and launched whenever the user logs in. This application notifies the user about various POA events such as update completion and reboot required.

The Dell Precision Optimizer installer package is also responsible for creating a software registry key that is used by the Dell Precision Optimizer modules. Following are the default paths:

<table>
<thead>
<tr>
<th>Table 1. Default paths</th>
<th>Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation folder:</td>
<td>C:\Program Files\Dell\PPO</td>
</tr>
<tr>
<td>Registry Path:</td>
<td>HKLM\Software\Dell\PPO</td>
</tr>
<tr>
<td>Runtime Data:</td>
<td>C:\ProgramData\Dell\PPO</td>
</tr>
</tbody>
</table>

The installation package copies some default profiles and policies to the installation folder.

**Topics:**

- Uninstall Dell Precision Optimizer

# Uninstall Dell Precision Optimizer

Dell Precision Optimizer can be uninstalled from the system using the following steps:

The uninstall command can be fetched from the registry by reading the value of the string UninstallString from the following location:

<table>
<thead>
<tr>
<th>Table 2. Uninstall command location</th>
<th>Command location</th>
</tr>
</thead>
<tbody>
<tr>
<td>For 64bit system</td>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node \Microsoft\Windows\CurrentVersion\Uninstall{D66A3355-FEA4-4F60-8BAF-D6CBEDB396D8}</td>
</tr>
<tr>
<td>For 32bit system</td>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows \CurrentVersion\Uninstall{D66A3355-FEA4-4F60-8BAF-D6CBEDB396D8}</td>
</tr>
</tbody>
</table>

A sample value for UninstallString is displayed:

"C:\Program Files (x86)\InstallShield Installation Information\{D66A3355-FEA4-4F60-8BAF-D6CBEDB396D8}\setup.exe" -runfromtemp -10x0007 -removeonly

**NOTE:** In this command, the value -10x0007 may be different for the system.

From a command prompt (run as administrator), executing the above command launches the uninstall process of the Dell Precision Optimizer application. The following command can be modified to run the uninstall silently by adding the following command:

-s -f1<full-path-of-iss>

If the silent response file (.iss file) is in C:\temp folder and its name is uninst.iss, then following command will run uninstall silently:
"C:\Program Files (x86)\InstallShield Installation Information\{D66A3355-FEA4-4F60-8BAF-D6CBEDB396D8}\setup.exe" -runfromtemp -10x0007 -removeonly -s -f1c:\temp\uninst.iss
The performance subsystem consists of the following modules:

- Policy Processing Engine (PPE)
- Profile Update Tool (profUpd.exe)
- Update Options Tool (upgradeOpt.exe)

Topics:

- Policy Processing Engine
- Profile update tool
- Update options tool

### Policy Processing Engine

Policy Processing Engine or PPE is implemented as a Windows Service which starts execution as soon as the machine boots up regardless of whether the user is logged in.

This module provides an interface which can be used to perform the following tasks:

- Activate or de-activate profiles
- Enumerate input and output parameters to allow new policies to be created
- Save and retrieve profiles and policies for machines, users or third-party Dell Precision Optimizer aware applications

### Profile update tool

This command line tool (profUpd.exe) is used by Dell Precision Optimizer UI to check and update the profiles from the Dell server. A local configuration file must be used to inform Dell Precision Optimizer the Dell server address and which protocol to use such as HTTP, HTTPS, or FTP.

**NOTE:** All profiles and policies on the server are digitally signed and stored encrypted using AES-256 algorithm.

### Update options tool

This command line tool (upgradeOpt.exe) is used by Dell Precision Optimizer UI to start the Dell support site URL in the default browser. The site shows the list of hardware upgrade options available for the specific system.
The analysis subsystem provides the ability to generate the following types of reports:

- System Analysis Reports
- Workload Analysis Reports
- CPU Intelligence Reports
- GPU Intelligence Reports
- System Diagnostic Reports
- Performance Notifications

**System Analysis Reports** provides the data collected by Dell Data Vault (DDV) application in an .XML format.

The **Workload Analysis** feature allows the user to analyze their workload.

**Topics:**

- System analysis reports
- Workload analysis
- CPU intelligence reports
- GPU intelligence reports
- System Diagnostics Report
- Performance Notifications
- Upgrade options

### System analysis reports

The user can enable or disable these reports using the Dell Precision Optimizer COM interface. This interface allows the user to configure how often System Analysis reports are generated to enumerate and read existing reports. The System Analysis Report .XML file contains the report data which is divided into <ddv_group> and <ddv_subgroup> elements. All data related to the same category are under the same group.

All thermistor 0 related data will be under DDV_GROUP called **Thermistor 0**.

### Report settings

**Enable system analysis**

**Table 3. Enable system analysis**

<table>
<thead>
<tr>
<th>Attribute Detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Enable or disable check box</td>
</tr>
<tr>
<td>Default:</td>
<td>Disabled</td>
</tr>
<tr>
<td>Description:</td>
<td>This setting allows the DDV subsystem to start data collection. When disabled, DDV is not active. After enabling this option, DDV reports are generated periodically until the setting is manually turned off. Any change in this category selection should cause all existing DDV raw data to be discarded.</td>
</tr>
</tbody>
</table>
Generate report

Table 4. Generate report

<table>
<thead>
<tr>
<th>Attribute Detail</th>
<th>Description</th>
</tr>
</thead>
</table>
| Type:            | Select one of the following options:  
|                  | • After 24 hours (Daily)  
|                  | • After 12 hours  
|                  | • After 8 hours  
|                  | • After 6 hours  
|                  | • After 4 hours  
| Default:         | Daily  
| Description:     | After enabling this option, DDV collects raw data and generates reports periodically. This setting controls how often the raw data is processed by DDV and converted into a new report. Any change in this category selection should cause all existing DDV raw data to be discarded. |

Enable data collection

Table 5. Enable data collection

<table>
<thead>
<tr>
<th>Attribute Detail</th>
<th>Description</th>
</tr>
</thead>
</table>
| Type:            | More than one category can be selected from the following:  
|                  | • Battery  
|                  | • Thermal  
|                  | • Fan  
|                  | • Processor  
|                  | • Memory  
|                  | • Storage  
|                  | • Network  
| Default:         | All check boxes are enabled  
| Description:     | This setting allows the user to control which categories of data will be displayed in the report. |

Workload analysis

Dell Precision Optimizer 5.0 and later allows the user to characterize their workloads and determine their resource usages. When the user initiates the analysis, Dell Precision Optimizer collects the system resource usage parameters such as CPU, memory, disk, and GPU.

CPU intelligence reports

Dell Precision Optimizer 5.0 and later allows the user to view enhanced Intel CPU information which includes processor information as well as live data for each logical processor. This data is displayed in the form of line graphs by Dell Precision Optimizer UI.

The UI uses the COM interface to get the following processor information:

• CPU Name  
• Number of Sockets  
• Number of Physical Cores  
• Hyper-Threading State (Enabled/Disabled)  
• L1 Cache Size (KB)  
• L2 Cache Size (KB)  
• L3 Cache Size (KB)  
• CPU Utilization per logical processor
GPU intelligence reports

For supported Nvidia and AMD GPU adapters only, Dell Precision Optimizer 5.0 and later allows the user to view enhanced GPU. This includes GPU adapter and software information as well as live data for each GPU. This data is displayed in the form of line graphs by Dell Precision Optimizer UI.

The UI will use Dell Precision Optimizer interface to get the following GPU information:

- Number of GPUs
- GPU Driver Version
- GPU Adapter Name (Active GPU 0 Only)
- Video BIOS Version (Active GPU 0 Only)
- Framebuffer Size (Active GPU 0 Only)

The GPU live data is available from certain Nvidia & AMD adapters only, when a user is logged in.

**NOTE:** On some mobile systems with AMD GPU adapters, valid live data may only be displayed when an active load is running on the AMD GPU adapter.

The following live information for each GPU shall be collected and displayed in the form of line graphs.

- GPU Utilization
- GPU Temperature
- GPU Fan #0 Speed (%)
- Video Memory Utilization

System Diagnostics Report

Dell Precision Optimizer 5.0 and later allows the user to run System Diagnostics reports. These are standard Microsoft-provided reports such as a System Report, Battery Report, and Reliability Report. The user must be able to generate a new report or view the last report generated previously. Only users with local administrator privileges can use this option.

This feature acts as a shortcut to existing Microsoft tools. The following reports are available from this dashboard:

- System Diagnostics Report contains:
  - Diagnostics results listing errors and warning in the system
  - Resource usage overview
- System Reliability Report contains:
  - List of Application, Windows & Miscellaneous failures in the last few weeks
  - Informational Events and Warnings during that period
  - Windows Stability Index
- Battery Report (This feature is available only on Windows 8 and later versions):
  - Installed Battery Details
  - Recent Usage and History
  - Battery Capacity and Life Estimates

Performance Notifications

Dell Precision Optimizer 4.0 and later allows you to enable Performance Notifications. It allows you to get notifications in any of the following cases:

- Excessive CPU utilization
- Excessive memory utilization
- Excessive disk read or write operations
Upgrade options

Dell Precision Optimizer UI provides a new link to Dell support website where a user may view and order optional or upgrade parts for a specific platform. The Service Tag of the system must be used by the application to determine what upgrades are available. This feature is internally implemented using the upgratdeOpt.exe tool.
Dell Precision Optimizer System maintenance or SM allows you to filter updates that are seen or applied based on the following criteria derived from Dell Command | Update:

- Criticality (Critical, Recommended, and Optional)
- Type (Hardware Drivers, Application, BIOS, and Firmware)
- Category (Audio, Chipset, Input, Network or Bluetooth, Storage, Video, and Others)
User feedback

Dell Precision Optimizer UI provides an option to the user to send feedback back to Dell. Dell Precision Optimizer UI provides a link or button which the user can click to initiate this feedback. UI launches a URL in the browser which will allow the user to use a Dell standard form to provide feedback for Dell Precision Optimizer.
Improve Dell Precision Optimizer

The Dell Precision Optimizer Customer Experience Improvement Program allows Dell customers to impact the development of future Dell Precision Optimizer releases. By sharing information with Dell regarding how you use Dell Precision Optimizer, you can contribute to improvements of future versions of the product.

The Dell Precision Optimizer Customer Experience Improvement Program adheres to all of the provisions of the Dell privacy policy. The data collected is limited to Dell Precision Optimizer usage and the workstation's Service Tag. No personal information of data is collected. You may opt in or out of the program at any time.

This feature is disabled by default.
WMI Providers

Dell Precision Optimizer 5.0 and later includes a WMI provider to allow access to the following information. Please refer to Appendix A for MOF descriptions. The following two files are part of the Dell Precision Optimizer package:

- Dell Precision Optimizer WMI Provider: dpoProv.mof
- Dell Precision Optimizer SMS MOF definition file: sms_def_dpo3.mof
- DDV Reports
- Product Version
- Last Check For Update Time
- Last System Update Time
- Last Check For Profiles
- Profile or Policy Trigger History
- List of Active Profiles
- Performance Notifications

DPOCMD.EXE

Dell Precision Optimizer 5.0 and later provides CLI tool, dpoCmd.exe, that allows the IT administrator the following capabilities:

- Add a new Profile or Policy
- List all Profiles
- Enable or Disable a Profile
- Schedule System Analysis report(s) with specific filters
- Run Dell Precision Optimizer Dell System Update with filters
- Check for Dell Precision Optimizer Dell System Updates with filters
- Export a user created profile
- Import a user created profile
- Upgrade to Premium version
- Enable/Disable UI features using the following CLI options; these are also controlled using new command line switches in Dell Precision Optimizer installer:

Table 6. Command line switches

<table>
<thead>
<tr>
<th>Control</th>
<th>Definition</th>
<th>Default</th>
<th>Command line switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProfileControl</td>
<td>If 0, then do not allow user to enable or disable profiles</td>
<td>1</td>
<td>PROFCTRL</td>
</tr>
<tr>
<td>ProfileUpdate</td>
<td>If 0, then do not allow user to check for new profiles</td>
<td>1</td>
<td>PROFUPD</td>
</tr>
<tr>
<td>SystemUpdate</td>
<td>If 0, then do not allow user to check for system updates</td>
<td>1</td>
<td>SYSUPD</td>
</tr>
<tr>
<td>DDVCControl</td>
<td>If 0, then do not allow user to enable or disable the System Analysis reports</td>
<td>1</td>
<td>DDVCTRL</td>
</tr>
<tr>
<td>UserFeedback</td>
<td>If 0, then do not allow user to send Dell Precision Optimizer feedback</td>
<td>1</td>
<td>USRFB</td>
</tr>
<tr>
<td>Control</td>
<td>Definition</td>
<td>Default</td>
<td>Command line switch</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------</td>
<td>---------</td>
<td>---------------------</td>
</tr>
<tr>
<td>UpgradeOptions</td>
<td>If 0, then do not allow user to check for upgrade options</td>
<td>1</td>
<td>UPGOPT</td>
</tr>
<tr>
<td>WorkloadAnalysis</td>
<td>If 0, then do not allow user to run Workload Analysis</td>
<td>1</td>
<td>WKLANL</td>
</tr>
<tr>
<td>GfxPlugins</td>
<td>If 0, then do not show GfxPlugin options to the user</td>
<td>1</td>
<td>GFXPLUGINS</td>
</tr>
<tr>
<td>ImproveDPO</td>
<td>If 0, then do not show user Improve Dell Precision Optimizer setting</td>
<td>1</td>
<td>IMPROVEDPO</td>
</tr>
<tr>
<td>ISVCertDrv</td>
<td>If 0, then do not allow user to view/install ISV Certified graphics drivers</td>
<td>1</td>
<td>ISVCERTGFX</td>
</tr>
<tr>
<td>SmartAlerts</td>
<td>If 0, then do not allow user to enable or disable the Smart Alerts</td>
<td>1</td>
<td>SMARTALERT</td>
</tr>
</tbody>
</table>

CLI Usage:

dpoCmd.exe -savePolicy <complete_dpx_path>
dpoCmd.exe -saveProfile <complete_dpx_path>
dpoCmd.exe -listProfiles
dpoCmd.exe -enableProfile <profile_guid>
dpoCmd.exe -disableProfile <profile_guid>
dpoCmd.exe -scheduleReports <numReports> <reportDuration> [-r <ddvSubSystem> ] [-r <ddvSubSystem>] ...

- where, <reportDuration> can be one of 0, 4, 6, 8, 12
  - 0 means daily report
  - 4 means 4 hour report
  - 6 means 6 hour report and so on.
- -r <ddvSubSystem> removes that subsystem and the data does not appear in the DDV reports that are generated.
  <ddvSubSystem> can be one of the following:
  - Battery
  - Thermal
  - Fan
  - Processor
  - Memory
  - Network
  - Storage

dpoCmd.exe –cancelReports

dpoCmd.exe -enableFeatures <feature> [<<feature> ...]. where <feature> can be one of the following:
  - PROFCTRL
  - PROFUPD
  - SYSUPD
  - DDVCTRL
  - USRFB
  - UPGOPT
  - WKLANL
  - GFXPLUGINS
  - IMPROVEDPO
  - ISVCERTGFX
  - SMARTALERT

dpoCmd.exe -disableFeatures <feature> [<<feature> ...]. where <feature> can be one of the following:  

· PROFCTRL
· PROFUPD
· SYSUPD
· DDVCTRL
· USRFB
· UPGOPT
· WKLANL
· GFXPLUGINS
· IMPROVEDPO
· ISVCERTGFX
· SMARTALERT

dpoCmd.exe -checkForUpdatesNow -criticality:CRO -filter:BDAF -device:ACMSNV <activityLogFileName>

where -criticality: can be one or more of the following:
- C => Critical
- R => Recommended
- O => Option

where -filter: can be one or more of the following:
- B => BIOS
- D => Drivers
- A => Applications
- F => Firmware

where -device: can be one or more of the following:
- A => Audio
- C => Chipset
- M => Mouse/Keyboard
- S => Storage
- N => Network/Bluetooth
- V => Video

dpoCmd.exe -exportProfile <profile_guid or unique_profile_name> <dpzFileName>
dpoCmd.exe -importProfile <dpzFileName>
dpoCmd.exe -upgradeToPremium <licenseKey>

where licenseKey is an alphanumeric key and not a file that contains the key.

**Setup command line switches**

Dell Precision Optimizer 5.0 installer provides command line switches to allow IT administrator to control certain behaviors of the client package. This list is mentioned in the Section 8.2.

Table 7. Setup command line switches

<table>
<thead>
<tr>
<th>Command line switches</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup.exe PROFUPD=0 WKLANL=0</td>
<td>To install Dell Precision Optimizer, where the user is not allowed to check for new profiles or run workload analysis.</td>
</tr>
</tbody>
</table>

In addition, a new option GUI=0, allows the IT administrator to install the Dell Precision Optimizer client without any UI component, that is headless mode. The user cannot control the software. The IT administrator may use the new CLI tool to enable or disable other runtime features.
SCCM

This is one of the methods used by the IT administrators today to centrally manage their systems and software applications. In this section, there are examples of how IT Administrators can use SCCM tools to manage the Dell Precision Optimizer application.

NOTE: There are methods and tools other than SCCM in the industry. Use the listed examples to manage Dell Precision Optimizer in those environments.

Instructions for creating the Dell Precision Optimizer application package

Follow these steps to create a Dell Precision Optimizer package that can be deployed on a selected client system in the Enterprise. Note: The steps may differ slightly based on the SCCM version that you are using.

1. Download the Dell Precision Optimizer files required for installation.
2. In the Configuration Manager Console:
   - Open the Software Library Page.
   - Click Overview folder.
   - Click Application Management.
   - Right click Applications and select Create Application.
3. In the Create Application wizard:
   - Select Manually and specify the application information.
   - Give the application a name, that is Dell Precision Optimizer 5.00.02, click Next.
   - Click Next in the Application Catalog.
   - Click Add in the Deployment Types page.
   - In the Create Deployment Type wizard, select Type: script installer, click Next.
   - Give the deployment type a name, click Next.
   - Type the location of the Dell Precision Optimizer files in the Content Location.
   - Type the following in Installation Program: “PoaInstaller.exe” /s
   - In the Detection Methods tab, click Add clause.
   - The detection rule is as follows:

```
Setting Type: Registry
Hive: HKEY_LOCAL_MACHINE
Key: Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{D66A3355-FEA4-4F60-8BAF-D6CBEDB396DB}
```
   - Click OK to close the Detection Rule window, click Next in the Create Deployment Type wizard.
   - Specify the user experience as follows:
     - Installation behavior: Install for system
     - Login Requirement: Whether or not a user is logged on
     - Installation Program Visibility: Normal
   - Click Next in the Requirements tab.
   - Click Next in the Dependencies tab.
   - Click Next in the Summary and verify that the Deployment Type is created successfully. Close the Create Deployment Type wizard.
4. In the Create Application wizard, click Next in the Deployment Types tab, click Next in the Summary tab and confirm that the application is created successfully.

Instructions for deploying application

After creating the package, use the following instruction to deploy it to selected clients:

1. Right click the application to deploy and select Deploy.
2. Select the device collection on which you would want to install Dell Precision Optimizer.
3. Ensure that the Automatically distribute content for dependencies option is checked, click Next.
4. In the Content tab, click Add to select the distribution point.
5. In the **Deployment settings** tab, have the following:
   - Action: Install
   - Purpose: Required

6. In the **Scheduling** tab, click **Next**.

7. In the **User Experience** tab, select User notifications: Display in Software Center, and only show notifications for computer restarts.

8. Click **Next** in the **Alerts** tab, click **Next** in the **Summary** tab, and verify deployment completion.

### Verify deployment success in client systems

To verify the successful deployment in the client systems:

1. Open **Software Center** in the client system, verify if the Dell Precision Optimizer is installed.

   **NOTE:** It may take a few minutes for the installation to take place after deploying the application

2. Go to `C:\Windows\CCM\Logs` and check `AppDiscovery.Log`, `AppIntent.log`, and `AppEnforce.log` for troubleshooting purpose.

### Changing the Dell Precision Optimizer Client behavior using DPOCMD.EXE

Following are the steps to run Dell Precision Optimizer CLI (`dpoCmd.exe`) on a target system to change the behavior of Dell Precision Optimizer software on that system. The following example illustrates the use of `dpoCmd.exe` to enable a Dell Precision Optimizer profile (After effects by Adobe).

#### Create a software package

To create a software package:

1. In the **Configuration Manager** Console:
   a. Open the **Software library** page.
   b. Click **Overview** tab.
   c. Open the **Application Management** tab.
   d. Right click **Packages** and select **Create new package**.

2. In the **Create Package and Program** wizard:
   a. Set the Name: Enable a Dell Precision Optimizer profile.
   b. Specify information about the package and click **Next**.

3. In the **Program Type** tab, select **Standard program**.

4. In the **Standard program** tab:
   a. Name: Enable Adobe after effects
   b. Command line: `dpoCmd.exe –enableProfile {2F066600-FA52-4F57-890D-2621D39B0BE9}`
   c. Startup folder: `C:\program files\dell\ppo`
   d. Run: Normal
   e. Program can run: Whether or not a user is logged in
   f. Run mode: Run with administrative rights
   g. Drive mode: Runs with UNC name

5. In the **Requirements** tab, select **This program can run on any platform**.

6. Click **Next**, review the package summary and verify the package was created successfully.

#### MAKE Deploy the enable a Dell Precision Optimizer profile software package

1. In the **Configuration Manager** console:
   a. Open the **Software library** page.
   b. Click the **Overview** tab.
   c. Open the **Application Management** tab.
d. Click Packages.
2. Right click the Enable DPO profile software package and select Deploy.
3. In the Deploy Software Wizard:
   a. In the General tab, click browse to select the device collection, click Next.
   b. In the Content tab, click Add to add a distribution point, click Next.
   c. In the deployment settings tab, have the following:
      - Action: Install
      - Purpose: Required
      - Check the Send wake-up packets box.
   d. In the Scheduling tab, select the time of deployment and ensure that the Rerun behavior option is set to Always rerun program. To deploy now, click New and select Assign immediately after this event: As soon as possible.
   e. In the User Experience tab ensure that the following check boxes are checked
      1. Software Installation
      2. System restart(if required to complete installation)
      3. Commit changes at deadline or during a maintenance window (requires restarts).
   f. In the Distribution points tab:
      1. Deployment options: Download content from distribution point and run locally.
      2. Ensure that the Allow clients to share content with other clients on the same subnet option is checked.
   g. Click Next and verify that the deployment is successfully completed.

SSRS Reports
As a system administrator, you can create various reports based on the data collected from Dell Precision Optimizer’s WMI providers. If this is desired, you can include the sms_def_dpo3.mof to extend the database definitions and fetch the corresponding data from Dell Precision Optimizer client systems. You may select some or all the data elements you want to review. The default is set to select all Dell Precision Optimizer data elements.

Importing the sms_def_dpo3.mof file to set hardware inventory classes
1. In the Configuration Manager console:
   a. Open the Administration page.
   b. Click the Overview tab.
   c. Click the Site Configuration tab and select Client Settings.
2. Right click an existing Client Setting and select the properties or create a new Custom Client Setting.
3. In the Hardware Inventory tab, select Set Classes.
4. Select Import and browse to the location of the sms_def_dpo3.mof file.
5. Click Ok to import the file and close the Hardware Inventory Classes window.

After the collected data is populated in the SQL database, you can create different type of Dell Precision Optimizer reports. The samples (*.RDL) are provided with the Dell Precision Optimizer software. You can import these RDL files, connect them to your SQL database and run the reports.

To Import an .RDL file
1. Open SQL Server data tools.
2. In the Solution Explorer, right click the folder in which you would like to add the .RDL file.
   a. Select Add Existing Item.
   b. Select the .RDL file.
3. Once the file is imported, open the file and select the Design tab.

To ensure the .RDL file is using the right data source
1. In the Report Data pane, click Datasets and right click one of the datasets and select Dataset Properties.
2. In the Dataset Properties window:
   a. the Use a dataset embedded in my report is selected.
   b. Under Data source, click New....
c. In the Data Source Properties window, select Use shared data source reference and select the correct data source.

d. Click Ok.

3. Repeat steps 1 and 2 for all other datasets in the Datasets folder.

Disk Information across multiple systems

This report displays disk information across multiple systems using their latest system analysis report.

Table 8. Disk information across multiple systems

<table>
<thead>
<tr>
<th>Service Tag</th>
<th>Bytes Read (MB)</th>
<th>Bytes Write (MB)</th>
<th>Read Time (%)</th>
<th>Write Time (%)</th>
<th>Idle Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>98K94X1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>F8W4WZ1</td>
<td>0</td>
<td>460</td>
<td>0</td>
<td>0</td>
<td>98</td>
</tr>
</tbody>
</table>

Disk information for a single system

This report displays disk information for a single system across multiple reports.

Table 9. Disk information for a single system

<table>
<thead>
<tr>
<th>Date Reported</th>
<th>Bytes Read (MB)</th>
<th>Bytes Write (MB)</th>
<th>Read Time (%)</th>
<th>Write Time (%)</th>
<th>Idle Time (%)</th>
<th>Hours On</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-07-15T11:06:53-05:00</td>
<td>5</td>
<td>526</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>2015-07-15T07:06:53-05:00</td>
<td>16</td>
<td>606</td>
<td>0</td>
<td>1</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>Date Reported</td>
<td>Bytes Read (MB)</td>
<td>Bytes Write (MB)</td>
<td>Read Time (%)</td>
<td>Write Time (%)</td>
<td>Idle Time (%)</td>
<td>Hours On</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------</td>
<td>------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>2015-07-15T03:00</td>
<td>568</td>
<td>949</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>3.1</td>
</tr>
<tr>
<td>2015-07-14T23:00</td>
<td>6</td>
<td>550</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>2015-07-14T19:07</td>
<td>0</td>
<td>466</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>4</td>
</tr>
<tr>
<td>2015-07-14T15:00</td>
<td>0</td>
<td>447</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>3.95</td>
</tr>
<tr>
<td>2015-06-22T15:2</td>
<td>369</td>
<td>354</td>
<td>0</td>
<td>0</td>
<td>96</td>
<td>3.48</td>
</tr>
<tr>
<td>2014-12-16T03:26</td>
<td>0</td>
<td>176</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>3.43</td>
</tr>
<tr>
<td>2014-12-15T23:26</td>
<td>0</td>
<td>178</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>4</td>
</tr>
<tr>
<td>2014-12-15T19:26</td>
<td>0</td>
<td>177</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>3.98</td>
</tr>
</tbody>
</table>

**KACE**

This is one of the methods used by the IT administrators today to centrally manage their systems and software applications. In this section, examples are provided of how IT Administrators can use KACE appliance to manage the Dell Precision Optimizer application.

**NOTE:** The following steps were verified on KACE appliance 6.4.120756 K1000. If you are using a different version of KACE, then the actual steps may vary slightly.
Instructions for deploying Dell Precision Optimizer using KACE

An IT administrator can use the following procedure to deploy Dell Precision Optimizer application on select client systems in their domain.

Create an installation script

From your KACE appliance console:

1. Navigate to **Scripting->Scripts->Choose Action->New.**
2. On the **Script Detail** page, enter the following information:
   - **Name** -> Install DPO
   - **Enabled** -> Check the box
   - **Type** -> Online K-Script
   - **Description** -> This script will install DPO client software
   - **Deploy** -> None
   - **Operating Systems** -> Clear the Select Specific Operating Systems and select Microsoft Windows
   - **(Alternate) Operating Systems** -> Select specific Windows OS for deployment
   - **Windows Run As** -> Local System
   - **Notify** -> None
   - **Schedule** -> None
   - **Dependencies** -> Add all DPO package files as new dependencies
   - **Tasks** -> Select New Task
   - **Verify** -> Click **Add**, then select **Launch a program**, enter the following data:
     - **Directory** -> $(KACE_DEPENDENCY_DIR)
     - **File** -> PoaInstaller.exe
     - **Wait for Completion** -> CHECKED
     - **Visible** -> UNCHECKED
     - **Parameters** -> LOGFILE=c:\temp\dpo.log /s
     - Save changes.
   - **On Success** -> None
   - **Remediation** -> None
   - **One Remediation Failure** -> None
   - **Tasks** -> Select New Task
   - **Verify** -> Click **Add**, then select **Verify a file exists**, enter the following data:
     - **Directory** : C:\Program Files\Dell\DPO
     - **File** : dpoCmd.exe
     - **Save Changes.**
   - **On Success** -> None
   - **Remediation** -> None
   - **One Remediation Failure** -> None
   - Click **Save**.

Run installation script on select systems

From your KACE appliance console:

1. Click **Scripting** and then select **Run Now.**
2. Select **Install DPO** from the Scripts drop down menu.
3. Under **Labels**, select a label of Windows devices where you wish to deploy Dell Precision Optimizer OR manually select a set of systems.
4. Click **Run Now.**
5. Click **Save.**

This initiates the deployment of Dell Precision Optimizer client software on selected systems. These steps can be customized as well.
Changing Dell Precision Optimizer Client Behavior using DPOCMD.EXE

Following are the steps to run Dell Precision Optimizer CLI (dpoCmd.exe) on a target system to change the behavior of Dell Precision Optimizer software on that system. The following example illustrates the use of dpoCmd.exe to enable a Dell Precision Optimizer profile (After Effects by Adobe).

Create an installation script

From your KACE appliance console:

2. On the Script Detail page, enter the following information:
   • Name -> Enable DPO profile after effects
   • Enabled -> Check the box
   • Type -> Online K-Script
   • Description -> This script will enable the After Effects profile under DPO client software
   • Deploy -> None
   • Operating Systems -> Clear the Select Specific Operating Systems and select Microsoft Windows
   • (Alternate) Operating Systems -> Select specific Windows OS for deployment
   • Windows Run As -> Local System
   • Notify -> None
   • Schedule -> None
   • Dependencies -> None
   • Tasks -> Select New Task
   • Verify -> Click Add, then select Launch a program, enter the following data:
     • Directory -> C:\Program Files\Dell\PPO
     • File -> dpoCmd.exe
     • Wait for Completion -> CHECKED
     • Visible -> UNCHECKED
     • Parameters -> -enableProfile {2F066600-FA52-4F57-890D-2621D39B0BE9}
     • Save changes.
   • On Success -> None
   • Remediation -> None
   • On Remediation Success -> None
   • One Remediation Failure -> None
   • Tasks -> Select New Task
   • Verify -> Click Add, then select Verify a file exists, enter the following data:
     • Directory -> C:\Program Files\Dell\DPO
     • File -> dpoCmd.exe
     • Save Changes.
   • On Success -> None
   • Remediation -> None
   • On Remediation Success -> None
   • One Remediation Failure -> None
   • Click Save.

Run this script on select systems

From your KACE appliance console:

1. Click Scripting and then select Run Now.
2. Select Enable DPO profile after effects from the Scripts drop down menu.
3. Under Labels, select a label of Windows devices where you wish to deploy Dell Precision Optimizer OR manually select a set of systems.
4. Click Run Now.
Custom reports

Here are some examples on how you can collect some data from Dell Precision Optimizer clients using its WMI classes and create custom reports. Dell Precision Optimizer provides a large set of WMI classes to allow an IT administrator to create a huge variety of reports. The following steps illustrate how to create a Dell Precision Optimizer report. An IT administrator can customize what data must be collected and know how to present that data.

Create Custom Inventory Rules

From your KACE appliance console:

1. Click Inventory, then select Software.
2. Choose Action and select New.
3. On the Software Details page, enter the following information
   - Name -> DPO Sample Inventory
   - Version -> v1
   - Publisher -> Dell
   - Supporting Operating Systems -> Select OSes
   - Custom Inventory Rule -> ShellCommandTextReturn(wmic /namespace:\root\cimv2\DPO Path DPO_Profiles get /ALL)
   - Click Save.
4. Click back into the new custom inventory record and hover over the record just created. Note the identifier(ID#) at the end of the URL. The URL with the ID# is visible at the lower left hand corner of the page. You will need this later for creating the report.

Force Inventory Collection

From your KACE appliance console:

1. Click Inventory and select Devices.
2. Select the device(s) where Dell Precision Optimizer is installed (you could use a SmartLabel for this purpose).
3. Choose Action and select Force Inventory.
4. After the inventory cycle is completed, navigate into one of the selected devices that was online.
5. On the Device Detail page, click Software, expand Custom Inventory Fields. This displays a list of profiles and their current states.

NOTE: Now that you have Script and Custom Inventory setup, and completed a Custom Inventory cycle on all desired systems, it is time to use K1000s reporting capabilities! While you can definitely pull the Dell Precision Optimizer information out of the K1000 using a Wizard based report, we are going to use a custom SQL report to process and filter our information into a useful report.

Create report

From your KACE appliance console:

1. Click Reporting, then select Reports.
2. Choose Action and select New (SQL).
3. On the Report Detail page, enter the following data:
   - Title -> Dell Precision Optimizer Sample Profile Report
   - Description -> This is a sample Dell Precision Optimizer report ...
   - Category <any> or New Category -> DPO Reports
   - SQL ->

```sql
SELECT
  MACHINE.NAME AS Name,
  MACHINE.IP AS Ip,
  MACHINE.USER_LOGGED AS LoggedUser,
  MACHINE.CS_MANUFACTURER AS Manufacturer,
  MACHINE.CS_MODEL AS Model,
  MACHINE_CUSTOM_INVENTORY.STR_FIELD_VALUE AS MACHINE_CUSTOM_INVENTORY_XXXX,
  COUNT(MACHINE_CUSTOM_INVENTORY.STR_FIELD_VALUE) AS Total_Devices
FROM
  MACHINE_CUSTOM_INVENTORY
JOIN
  MACHINE
ON
  MACHINE.ID = MACHINE_CUSTOM_INVENTORY.ID
WHERE
  MACHINE_CUSTOM_INVENTORY.SOFTWARE_ID = XXXX
GROUP BY
  MACHINE_CUSTOM_INVENTORY.STR_FIELD_VALUE
ORDER BY
  MACHINE.CS_MANUFACTURER ASC, MACHINE.CS_MODEL ASC
```
Replace XXXX with the ID# of your custom inventory which was collected above when the Custom Inventory Rule is created.

4. Click **Save**.

**Run Report**

From your KACE appliance console:

1. Click **Reporting**, then select **Reports**.
2. Search for **DPO** to view your reports.
3. Select the desired report, such as DPO Sample Profile Report, and select the report format you want, say HTML.
typedef enum {
    EXIT_CODE_SUCCESS = (int) 0,
    EXIT_CODE_ERROR_GET_COMP_NAME = (int) 1,
    EXIT_CODE_COINIT_FAILED = (int) 2,
    EXIT_CODE_PROFILE_NOT_FOUND = (int) 3,
    EXIT_CODE_ERROR = (int) 4,
    EXIT_CODE_ERROR_GET_COMP_SID = (int) 5,
    EXIT_CODE_COINIT_SECURITY_FAILED = (int) 6,
    EXIT_CODE_MISSING_COM_INTERFACE = (int) 7,
    EXIT_CODE_PROFILE_GETSTATE_FAILED = (int) 8,
    EXIT_CODE_PROFILE_SETSTATE_FAILED = (int) 9,
    EXIT_CODE_MISSING_STORE = (int) 40,
    EXIT_CODE_NULL_STORE = (int) 41,
    EXIT_CODE_READFILE_FAILED = (int) 42,
    EXIT_CODE_WRITEFILE_FAILED = (int) 43,
    EXIT_CODE_OUT_OF_MEM = (int) 44,
    EXIT_CODE_SAVE_STORE_FAILED = (int) 45,
    EXIT_CODE_ENCRYPTION_FAILED = (int) 46,
    EXIT_CODE_DDV_REPORTS_ALREADY_SCHEDULED = (int) 60,
    EXIT_CODE_ENABLE_DDV_FAILED = (int) 61,
    EXIT_CODE_SET_DDV_FILTERS_FAILED = (int) 62,
    EXIT_CODE_INVALID_DDV_REPORT_DURATION = (int) 63,
    EXIT_CODE_SET_REPORT_FREQ_FAILED = (int) 64,
    EXIT_CODE_SET_REPORT_NUM_FAILED = (int) 65,
    EXIT_CODE_DISABLE_DDV_FAILED = (int) 66,
    EXIT_CODE_ERROR_ENUM_DDV_SUBSYSTEMS = (int) 67,
    EXIT_CODE_DO_UPDATE_FAILED = (int) 70,
    EXIT_CODE_PREV_CHECK_FAILED = (int) 71,
    EXIT_CODE_PREV_UPDATE_ACTION_IN_PROGRESS = (int) 72,
    EXIT_CODE_REGISTER_EVENTS_FAILED = (int) 73,
    EXIT_CODE_CHECK_UPDATE_FAILED = (int) 74,
    EXIT_CODE_SET_FEATURE_FAILED = (int) 80,
    EXIT_CODE_UI_IS_RUNNING = (int) 98,
    EXIT_CODE_USAGE_ERROR = (int) 99
} EXIT_CODE;
#pragma autorecover

#pragma namespace("\\\\\逃离cimv2")
instance of __Namespace
{
  Name = "DPO" ;
};

#pragma namespace("\\\\\逃离cimv2\DPO")
/**
* There is one instance of this class for each summary
* file present on the system.
* The instance will contain all the hardware data and
* the statistics from the summary file.
* HardwareInfoGUID is the unique ID from the summary file.
* HardwareInfoGUID associates this instance with
* with instances of other dependent classes that may have
* multiple instances (e.g. DPO_Monitor. DPO_BiosInternalLogs etc.)
*/
[Description("An instance of this class contains all the hardware data and "
  statistics from a summary file.")],
Dynamic,Provider("DPOProv")
class DPO_HardwareInfo
{
  [Description("Unique ID from the summary file.")],
  Key
  string             HardwareInfoGUID;
  [Description("Revision of Dell Data Vault.")] string      DDV_Revision;
  [Description("Date/time when the summary file was created.")] string       File_Creation_Datetime;
  [Description("Date/time when Dell Data Vault began collecting the raw data.")] string     Data_Begining_Date;
  [Description("Date/time when Dell Data Vault stopped collecting the raw data and generated
  the statistics.")] string   Data_Ending_Date;
  [Description("Indicates whether this summary was created on service startup, regular timer or
  on demand.")]
  string             Summary_Type;
  [Description("Service Tag of the system obtained from the BIOS.")]
  string             System_Service_Tag;
  [Description("Customer Name 1")]
  string             Customer_Name_1;
  [Description("Customer Name 2")]
  string             Customer_Name_2;
  [Description("Customer Name 3")]
  string             Customer_Name_3;
  [Description("Customer specific data 1")]
  string             Customer_DEFINED_1;
  [Description("Customer specific data 2")]
  string             Customer_DEFINED_2;
  [Description("Customer specific data 3")]
  string             Customer_DEFINED_3;
  [Description("System Model")]
  string             System_Model;
  [Description("ePPID of the motherboard obtained from the BIOS.")]
string Motherboard_ePPID;
[Description("Current BIOS Version.")]
string BIOS_Version;
[Description("Type of the system eg. Laptop or Desktop")]
string System_Type;
[Description("Serial number of the CPU.")]
string Processor_Serial_Number;
[Description("Processor name.")]
string Processor_Name;
[Description("Processor speed.")]
string Processor_Speed;
[Description("Average of the percentage LCD brightness when the system was on AC.")]
sint16 LCD_Avg_Brightness_AC_Pct;
[Description("Average of the percentage LCD brightness when the system was on battery.")]
sint16 LCD_Avg_Brightness_DC_Pct;
[Description("Video Controller name.")]
string Video_Controller;
[Description("Video controller memory size.")]
sint32 Video_RAM_Bytes;
[Description("Number of displays on the system.")]
sint16 Number_of_Displays;
[Description("Operating system, 32bit vs 64bit & system locale information.")]
string Operating_System;
[Description("AC adapter power (for notebooks only).")]
string AC_Adapter_Type_W;
[Description("Number of hours the system was on.")]
real32 Hours_On;
[Description("Number of hours the system was on when powered by AC.")]
real32 Hours_On_AC;
[Description("Number of hours the system was on when powered by battery (for notebooks only).")]
real32 Hours_On_Batt;
[Description("Number of times the AC adapter was inserted in the system (for notebooks only).")]
sint16 No_Of_AC_Insertions;
[Description("Number of times the primary battery was inserted into the system (for notebooks only).")]
// NameChange sint16 Number_Of_Battery_Insertions;
[Description("Number of battery sessions where the session was between 0 to 30 mins (for notebooks only).")]
sint16 Battery_Sessions_0_30mins;
[Description("Number of battery sessions where the session was between 30 mins to 1 hr (for notebooks only).")]
sint16 Battery_Sessions_30min_1hr;
[Description("Number of battery sessions where the session was between 1 to 2 hrs (for notebooks only).")]
sint16 Battery_Sessions_1_2hr;
[Description("Number of battery sessions where the session was between 2 to 3 hrs (for notebooks only).")]
sint16 Battery_Sessions_2_3hr;
[Description("Number of battery sessions where the session was between 3 to 4 hrs (for notebooks only).")]
sint16 Battery_Sessions_3_4hr;
[Description("Number of battery sessions where the session was between 4 to 6 hrs (for notebooks only).")]
sint16 Battery_Sessions_4_6hr;
[Description("Number of battery sessions where the session was between 6 to 8 hrs (for notebooks only).")]
sint16 Battery_Sessions_6_8hr;
[Description("Number of battery sessions where the session was between 8 to 12 hrs (for notebooks only).")]
sint16 Battery_Sessions_8_12hr;
[Description("Number of battery sessions where the session was greater than 12 hrs (for notebooks only).")]
sint16 Battery_Sessions_GT12hr;
[Description("Number of times the system entered Hibernate state (S4).")]
sint16 S4_Requests;
[Description("Total time the system was in Hibernate state (S4).")]
sint16 S4_Requests;
real32 S4_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 0 to 30 mins.")] sint16 S4_Event_Bin_0_30_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 30 mins to 1 hr.")] sint16 S4_Event_Bin_30_60_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 1 hr to 2 hrs.")] sint16 S4_Event_Bin_60_120_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 2 to 4 hrs.")] sint16 S4_Event_Bin_120_240_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 4 to 8 hrs.")] sint16 S4_Event_Bin_240_480_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was between 8 to 16 hrs.")] sint16 S4_Event_Bin_480_960_mins;
[Description("Number of times the system was in Hibernate state (S4) where the time in S4 was greater than 16 hrs.")] sint16 S4_Event_Bin_GT_960_mins;
[Description("Number of times the system entered Standby/Sleep state (S3)." )]
real32 S3_mins;
[Description("Total time the system was in Standby/Sleep state (S3)." )]
sint16 S3_Requests;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 0 to 30 mins." )]
sint16 S3_Event_Bin_0_30_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 30 mins to 1 hr." )]
sint16 S3_Event_Bin_30_60_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 1 hr to 2 hrs." )]
sint16 S3_Event_Bin_60_120_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 2 to 4 hrs." )]
sint16 S3_Event_Bin_120_240_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 4 to 8 hrs." )]
sint16 S3_Event_Bin_240_480_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was between 8 to 16 hrs." )]
sint16 S3_Event_Bin_480_960_mins;
[Description("Number of times the system was in Standby/Sleep state (S3) where the time in S3 was greater than 16 hrs." )]
sint16 S3_Event_Bin_GT_960_mins;
[Description("Average CPU consumption for all processors combined." )]
real32 Avg_CPU_Consumption;
[Description("Number of times the CPU consumption was 0%." )]
sint16 CPU_0_Pct;
[Description("Number of times the CPU consumption was between 0 to 20%." )]
sint16 CPU_1_20_Pct;
[Description("Number of times the CPU consumption was between 20 to 40%." )]
sint16 CPU_20_40_Pct;
[Description("Number of times the CPU consumption was between 40 to 60%." )]
sint16 CPU_40_60_Pct;
[Description("Number of times the CPU consumption was between 60 to 80%." )]
sint16 CPU_60_80_Pct;
[Description("Number of times the CPU consumption was between 80 to 100%." )]
sint16 CPU_80_100_Pct;
[Description("Average CPU throttle (for all processors combined)." )]
real32 Avg_CPU_Throttle;
[Description("Number of times the CPU throttle was 0%." )]
sint16 Throttle_0_Pct;
[Description("Number of times the CPU throttle was between 0 to 25%." )]
sint16 Throttle_1_25_Pct;
[Description("Number of times the CPU throttle was between 25 to 50%." )]
sint16 Throttle_25_50_Pct;
[Description("Number of times the CPU throttle was between 50 to 75%." )]
sint16 Throttle_50_75_Pct;
[Description("Number of times the CPU throttle was between 75 to 100%." )]
sint16 Throttle_75_100_Pct;
[Description("Percentage of time the processor (all processors combined) was in C1 state.")]
sint16 C1_State_Pct;
[Description("Percentage of time the processor (all processors combined) was in C2 state.")]
sint16 C2_State_Pct;
[Description("Percentage of time the processor (all processors combined) was in C3 state.")]
sint16 C3_State_Pct;
[Description("Percentage of time the processor (all processors combined) was in C0 state.")]
sint16 C0_State_Pct;
[Description("Number of LID transitions. One open-close is considered as one transition.")]
sint16 Lid_Transitions;
[Description("Number of hours the system was ON with LID open.")]
real32 Lid_Hours_Open;
[Description("Number of hours the system was ON with LID closed.")]
real32 Lid_Hours_Closed;
[Description("Number of dock events.")]
sint16 Number_Dock_Events;
[Description("Total system RAM memory.")]
string System_RAM_Bytes;
[Description("Total system RAM memory in GB.")]
real32 System_RAM_GB;
[Description("Percentage of time the system had to access hard disk to resolve page faults.")]
sint16 pgs_per_sec_pct;
[Description("Minimum number of pages read from or written to the disk to resolve hard page faults.")]
sint32 min_pgs_per_sec;
[Description("Maximum number of pages read from or written to the disk to resolve hard page faults.")]
sint32 max_pgs_per_sec;
[Description("Average number of pages read from or written to the disk to resolve hard page faults.")]
real32 avg_pgs_per_sec;
[Description("Percentage of time the system had between 0 to 256 MB of free physical memory.")]
real32 FreeMem_0_256MB_Pct;
[Description("Percentage of time the system had between 256 MB to 512 MB of free physical memory.")]
real32 FreeMem_256_512MB_Pct;
[Description("Percentage of time the system had between 512 MB to 768 MB of free physical memory.")]
real32 FreeMem_512_768MB_Pct;
[Description("Percentage of time the system had between 768 MB to 1024 MB of free physical memory.")]
real32 FreeMem_768_1024MB_Pct;
[Description("Percentage of time the system had between 1024 MB to 1280 MB of free physical memory.")]
real32 FreeMem_1024_1280MB_Pct;
[Description("Percentage of time the system had between 1280 MB to 1536 MB of free physical memory.")]
real32 FreeMem_1280_1536MB_Pct;
[Description("Percentage of time the system had between 1536 MB to 1792 MB of free physical memory.")]
real32 FreeMem_1536_1792MB_Pct;
[Description("Percentage of time the system had between 1792 MB to 2048 MB of free physical memory.")]
real32 FreeMem_1792_2048MB_Pct;
[Description("Percentage of time the system had between 2048 MB to 2304 MB of free physical memory.")]
real32 FreeMem_2048_2304MB_Pct;
[Description("Percentage of time the system had between 2304 MB to 2560 MB of free physical memory.")]
real32 FreeMem_2304_2560MB_Pct;
[Description("Percentage of time the system had between 2560 MB to 2816 MB of free physical memory.")]
real32 FreeMem_2560_2816MB_Pct;
[Description("Percentage of time the system had between 2816 MB to 3072 MB of free physical memory.")]
real32 FreeMem_2816_3072MB_Pct;
[Description("Percentage of time the system had more than 3072 MB of free physical memory.")]
real32 FreeMem_GT3072MB_Pct;
[Description("Percentage of time the system had between 0 to 256 MB of physical memory available to processes running on the computer.")]
real32 AvailMem_0_256MB_Pct;
real32                  AvailMem_256_512MB_Pct;
[Description("Percentage of time the system had between 256 MB to 512 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_512_768MB_Pct;
[Description("Percentage of time the system had between 768 MB to 1024 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_768_1024MB_Pct;
[Description("Percentage of time the system had between 1024 MB to 1280 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_1024_1280MB_Pct;
[Description("Percentage of time the system had between 1280 MB to 1536 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_1536_1792MB_Pct;
[Description("Percentage of time the system had between 1536 MB to 1792 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_1792_2048MB_Pct;
[Description("Percentage of time the system had between 1792 MB to 2048 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_2048_2304MB_Pct;
[Description("Percentage of time the system had between 2048 MB to 2304 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_2304_2560MB_Pct;
[Description("Percentage of time the system had between 2304 MB to 2560 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_2560_2816MB_Pct;
[Description("Percentage of time the system had between 2560 MB to 2816 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_2816_3072MB_Pct;
[Description("Percentage of time the system had between 2816 MB to 3072 MB of physical memory
available to processes running on the computer.")]

real32                  AvailMem_GT3072MB_Pct;
[Description("Percentage of time the system had more than 3072 MB of physical memory
available to processes running on the computer.")]

real32                  Average_PQL;  
[Description("Average Processor Queue Length.")]
sint16                  Min_PQL;
[Description("Minimum Processor Queue Length.")]
sint16                  Max_PQL;
[Description("Maximum Processor Queue Length.")]
real32                  PQL_0_Pct;
[Description("Percentage of time the system has PQL = 0.")]
real32                  PQL_1_Pct;
[Description("Percentage of time the system has PQL = 1.")]
real32                  PQL_2_Pct;
[Description("Percentage of time the system has PQL = 2.")]
real32                  PQL_3_Pct;
[Description("Percentage of time the system has PQL = 3.")]
real32                  PQL_4_Pct;
[Description("Percentage of time the system has PQL = 4.")]
real32                  PQL_5_Pct;
[Description("Percentage of time the system has PQL = 5.")]
real32                  PQL_5_10_Pct;
[Description("Percentage of time the system has PQL between 5 and 10.")]
real32                  PQL_10_20_Pct;
[Description("Percentage of time the system has PQL between 10 and 20.")]
real32                  PQL_GT20_Pct;
[Description("Average value of total system thread count.")]
real32                  Average_ThreadCount;
[sint64                  Min_ThreadCount;
[Description("Minimum value of total system thread count.")]
sint64                  Max_ThreadCount;
[Description("Maximum value of total system thread count.")]
sint64                  Std_Dev_ThreadCount;  
[Implemented]
void DeleteInstance ()
);  
************************************************************************************
* DPO_Monitor
* This has the monitor information from a summary log. There
* may be multiple instances of this class for each summary file.

APPENDIX B - WMI Class Definition Files   33
Monitor information from the summary log file. This information is extracted from the EDID data
"in the registry",
Dynamic, Provider("DPOProv")

```csharp
class DPO_Monitor
{
    [Description("Unique ID from the summary file.")]
    Key
    string HardwareInfoGUID;
    [
        Description("Monitor index number, starting from 0.")
    ]
    Key
    sint16 Index;
    [Description("Type of monitor (Dell or Non-Dell).")]
    string Monitor_Type;
    [Description("Model name of the monitor.")]
    string Model_Name;
    [Description("Serial number of the monitor.")]
    string Serial;
    [Description("Any vendor specific monitor data.")]
    string Vendor_Specific_Data;
};
```

DPO_HardwareInfoToMonitor

This class associates DPO_Monitor instance(s) with an instance of DPO_HardwareInfo.

```csharp
class DPO_HardwareInfoToMonitor
{
    [key] DPO_HardwareInfo REF Antecedent;
    [key] DPO_Monitor REF Dependent;
};
```

DPO_BiosInternalLogs

BIOS logs: All system logs such as BIOS, Diagnostics, IPMI, SMBIOS, SPD logs etc.

```csharp
class DPO_BiosInternalLogs
{
    [key] string HardwareInfoGUID;
    [key] string Name;
    string Time;
    [key] string LogType;
    string EventCode;
    string Descr;
};
```

DPO_HardwareInfoToBiosInternalLogs

This class associates DPO_HardwareInfoToBiosInternalLogs instance(s) with an instance of DPO_HardwareInfo.

```csharp
class DPO_HardwareInfoToBiosInternalLogs
{
    Association : ToInstance,
    [Association : ToInstance,
    Description("This class associates DPO_Monitor instance(s) with an instance of DPO_HardwareInfo.")]
    dynamic : ToInstance,
    PROVIDER("DPOProv") : ToInstance
}
```
class DPO_HardwareInfoToBiosInternalLogs {
  [key] DPO_HardwareInfo          REF        Antecedent;
  [key] DPO_BiosInternalLogs      REF        Dependent;
};

class DPO_WWAN {
  [Description("Unique ID from the summary file.")],
  Key
} string HardwareInfoGUID;

  [Description("Device name.")]
  string Device_Name;

  [Description("IMEI number.")]
  string IMEI;
};

class DPO_HardwareInfoToWWAN {
  [key] DPO_HardwareInfo          REF         Antecedent;
  [key] DPO_WWAN                  REF         Dependent;
};

class DPO_Battery {
  [Description("Unique ID from the summary file.")],
  Key
} string HardwareInfoGUID;

  [Description("Index number of the battery device starting from 1.")],
  Key
}sint16                Index;

  [Description("Manufacture date.")]
  string Manufacture_Date;

  [Description("Serial number.")]
  string Serial_Number;

  [Description("Chemistry.")]
  string Chemistry;

  [Description("Design Capacity in mAH.")]
  string Design_Capacity_mAH;

  [Description("Battery name.")]
  string Battery_Name;
};
string Name;
[Description("Manufacturer's name.")]
string Mfg_Name;
[Description("Full charge capacity of the battery.")]
sint32 FullChargeCapacity;
[Description("Battery cycle count.")]
sint16 Cycle_Count;
[Description("Total time (in minutes) the battery was in discharge state.")]
real32 Discharge_Time_mins;
[Description("Number of times the discharge depth was between 0 to 5\%.")]
sint16 Discharge_Depth_0_5_Pct;
[Description("Number of times the discharge depth was between 5 to 10\%.")]
sint16 Discharge_Depth_5_10_Pct;
[Description("Number of times the discharge depth was between 10 to 20\%.")]
sint16 Discharge_Depth_10_20_Pct;
[Description("Number of times the discharge depth was between 20 to 40\%.")]
sint16 Discharge_Depth_20_40_Pct;
[Description("Number of times the discharge depth was between 40 to 60\%.")]
sint16 Discharge_Depth_40_60_Pct;
[Description("Number of times the discharge depth was between 60 to 80\%.")]
sint16 Discharge_Depth_60_80_Pct;
[Description("Number of times the discharge depth was between 80 to 100\%.")]
sint16 Discharge_Depth_80_100_Pct;
[Description("Number of times the start of discharge was between 100 to 94\%.")]
sint16 Discharge_Start_Point_100_94_Pct;
[Description("Number of times the start of discharge was between 94 to 70\%.")]
sint16 Discharge_Start_Point_94_70_Pct;
[Description("Number of times the start of discharge was between 70 to 50\%.")]
sint16 Discharge_Start_Point_70_50_Pct;
[Description("Number of times the start of discharge was between 50 to 30\%.")]
sint16 Discharge_Start_Point_50_30_Pct;
[Description("Number of times the start of discharge was between 30 to 10\%.")]
sint16 Discharge_Start_Point_30_10_Pct;
[Description("Number of times the start of discharge was between 10 to 0\%.")]
sint16 Discharge_Start_Point_0_10_Pct;
[Description("Number discharge sessions where final RSOC was less than 15\%.")]
sint16 Discharge_Sess_End_10_15;
[Description("Number discharge sessions where final RSOC was less than 10\%.")]
sint16 Discharge_Sess_End_5_10;
[Description("Number discharge sessions where final RSOC was less than 5\%.")]
sint16 Discharge_Sess_End_LT_5;
[Description("Average temperature during battery discharge.")]
real32 Discharge_Temp_Avg;
[Description("Standard deviation of temperature during battery discharge.")]
real32 Discharge_Temp_Std_Dev;
[Description("Maximum temperature during battery discharge.")]
sint32 Discharge_Temp_Max;
[Description("Minimum temperature during battery discharge.")]
sint32 Discharge_Temp_Min;
[Description("Average current (in mA) during battery discharge.")]
real32 Discharge_mA_Avg;
[Description("Standard deviation of current (in mA) during battery discharge.")]
real32 Discharge_mA_Std_Dev;
[Description("Maximum current (in mA) during battery discharge.")]
sint32 Discharge_mA_Max;
[Description("Minimum current (in mA) during battery discharge.")]
sint32 Discharge_mA_Min;
[Description("Average voltage (in mV) during battery discharge.")]
real32 Discharge_mV_Avg;
[Description("Standard deviation of voltage (in mV) during battery discharge.")]
real32 Discharge_mV_Std_Dev;
[Description("Maximum voltage (in mV) during battery discharge.")]
sint32 Discharge_mV_Max;
[Description("Minimum voltage (in mV) during battery discharge.")]

sint32  Discharge_mV_Min ;
[Description("Average power (in W) during battery discharge.")]
real32  Discharge_Power_W_Avg;
[Description("Standard deviation of power (in W) during battery discharge.")]
real32  Discharge_Power_W_Std_Dev;
[Description("Maximum power (in W) during battery discharge.")]
sint32  Discharge_Power_W_Max;
[Description("Minimum power (in W) during battery discharge.")]
sint32  Discharge_Power_W_Min;
[Description("Percentage of time the power during discharge was between 0 to 5W.")]
sint16  Discharge_Power_0_5W_Pct;
[Description("Percentage of time the power during discharge was between 5 to 10W.")]
sint16  Discharge_Power_5_10W_Pct;
[Description("Percentage of time the power during discharge was between 10 to 15W.")]
sint16  Discharge_Power_10_15W_Pct;
[Description("Percentage of time the power during discharge was between 15 to 20W.")]
sint16  Discharge_Power_15_20W_Pct;
[Description("Percentage of time the power during discharge was between 20 to 25W.")]
sint16  Discharge_Power_20_25W_Pct;
[Description("Percentage of time the power during discharge was between 25 to 30W.")]
sint16  Discharge_Power_25_30W_Pct;
[Description("Percentage of time the power during discharge was between 30 to 40W.")]
sint16  Discharge_Power_30_40W_Pct;
[Description("Percentage of time the power during discharge was between 40 to 50W.")]
sint16  Discharge_Power_40_50W_Pct;
[Description("Percentage of time the power during discharge was between 50 to 60W.")]
sint16  Discharge_Power_50_60W_Pct;
[Description("Percentage of time the power during discharge was more than 60W.")]
sint16  Discharge_Power_GT60W_Pct;
[Description("Total time (in minutes) the battery was in charge state.")]
real32  Charge_Time_mins;
[Description("Number of sessions where the battery got fully charged.")]
// NameChange sint16  Charge_Number_Full_Charge_Sessions;
sint16  Num_Full_Charge_Sessions;
[Description("Number of sessions where the battery got partially charged.")]
// NameChange sint16  Charge_Number_Partial_Charge_Sessions;
sint16  Num_Partial_Charge_Sessions;
[Description("Average temperature during battery charge.")]
real32  Charge_Temp_Avg;
[Description("Standard deviation of temperature during battery charge.")]
sint16  Charge_Temp_Std_Dev;
[Description("Maximum temperature during battery charge.")]
sint16  Charge_Temp_Max;
[Description("Minimum temperature during battery charge.")]
sint16  Charge_Temp_Min;
[Description("Average current (in mA) during battery charge.")]
real32  Charge_mA_Avg;
[Description("Standard deviation of current (in mA) during battery charge.")]
real32  Charge_mA_Std_Dev;
[Description("Maximum current (in mA) during battery charge.")]
sint32  Charge_mA_Max;
[Description("Minimum current (in mA) during battery charge.")]
sint32  Charge_mA_Min;
[Description("Average voltage (in mV) during battery charge.")]
real32  Charge_mV_Avg;
[Description("Standard deviation of voltage (in mV) during battery charge.")]
real32  Charge_mV_Std_Dev;
[Description("Maximum voltage (in mV) during battery charge.")]
sint32  Charge_mV_Max;
[Description("Minimum voltage (in mV) during battery charge.")]
sint32  Charge_mV_Min;
[Description("Average power (in W) during battery charge when RSOC was less than 60%.")]

// NameChange real32    Charge_Power_W_RSOC_LE_60_Avg;
    real32    Charge_Pwr_RSOC_LE_60_Avg;
[Description("Standard deviation of power (in W) during battery charge when RSOC was less than 60%.")]
// NameChange real32    Charge_Power_W_RSOC_LE_60_Std_Dev;
    real32    Charge_Pwr_RSOC_LE_60_StdDv;
[Description("Maximum power (in W) during battery charge when RSOC was less than 60%.")]
// NameChange sint16    Charge_Power_W_RSOC_LE_60_Max;
    sint16    Charge_Pwr_RSOC_LE_60_Max;
[Description("Minimum power (in W) during battery charge when RSOC was less than 60%.")]
// NameChange sint16    Charge_Power_W_RSOC_LE_60_Min;
    sint16    Charge_Pwr_RSOC_LE_60_Min;
[Description("Average power (in W) during battery charge when RSOC was more than 60%.")]
// NameChange real32    Charge_Power_W_RSOC_LGT_60_Avg;
    real32    Charge_Pwr_RSOC_LGT_60_Avg;
[Description("Standard deviation of power (in W) during battery charge when RSOC was more than 60%.")]
// NameChange real32    Charge_Power_W_RSOC_LGT_60_Std_Dev;
    real32    Charge_Pwr_RSOC_LGT_60_StdDv;
[Description("Maximum power (in W) during battery charge when RSOC was more than 60%.")]
// NameChange sint16    Charge_Power_W_RSOC_LGT_60_Max;
    sint16    Charge_Pwr_RSOC_LGT_60_Max;
[Description("Minimum power (in W) during battery charge when RSOC was more than 60%.")]
// NameChange sint16    Charge_Power_W_RSOC_LGT_60_Min;
    sint16    Charge_Pwr_RSOC_LGT_60_Min;
[Description("Total time (in minutes) the battery was in dwell state.")]
    real32    Dwell_Time_mins;
[Description("Average RSOC level when the battery was in dwell state.")]
    real32    Dwell_Avg_RSOC_Level;
[Description("Average temperature during battery dwell state.")]
    real32    Dwell_Temp_Avg;
[Description("Standard deviation of temperature during battery dwell state.")]
    real32    Dwell_Temp_Std_Dev;
[Description("Maximum temperature during battery dwell state.")]
    sint32    Dwell_Temp_Max;
[Description("Minimum temperature during battery dwell state.")]
    sint32    Dwell_Temp_Min;
};

 *********************************************************************************************
* DPO_HardwareInfoToBattery
* This class associates DPO_Battery instance(s) with an instance of DPO_HardwareInfo.
*********************************************************************************************

Association : ToInstance,
Description(" This class associates DPO_Battery instance(s) with an" " instance of DPO_HardwareInfo."),
dynamic:ToInstance, PROVIDER("DPOProv") : ToInstance
}
class DPO_HardwareInfoToBattery
{
[key] DPO_HardwareInfo REF Antecedent;
[key] DPO_Battery REF Dependent;
};

 *********************************************************************************************
* DPO_NBFan
* This has the notebook fan information from a summary log. There may be multiple instances of this class for each summary file.
*********************************************************************************************
[Description("Notebook fan speed statistics.")], Dynamic,Provider("DPOProv") ]
class DPO_NBFan

APPENDIX B - WMI Class Definition Files
APPENDIX B - WMI Class Definition Files

```csharp
{ 
  [Description("Unique ID from the summary file.")],
  Key
}string    HardwareInfoGUID;

{ 
  [Description("Notebook fan index number starting from 0.")],
  Key
}sint16    Index;

[Description("Location where the fan is present in the system.")]
string    Location;

[Description("Percentage of time fan rpm was non-zero.")]
sint16    Fan_Duty_Cycle_Pct;

[Description("Fan speed when the summary log was generated.")]
sint32    RPM;

[Description("Peak fan speed.")]
sint32    Peak_Fan_RPM;

[Description("Average fan speed.")]
real32    Average_Fan_RPM;

[Description("Percentage of time the fan speed was 0 RPM.")]
sint16    RPM_0_Pct;

[Description("Percentage of time the fan speed was between 0 and 1000 RPMs.")]
sint16    RPM_0_1000_Pct;

[Description("Percentage of time the fan speed was between 1000 and 1700 RPMs.")]
sint16    RPM_1000_1700_Pct;

[Description("Percentage of time the fan speed was between 1700 and 2200 RPMs.")]
sint16    RPM_1700_2200_Pct;

[Description("Percentage of time the fan speed was between 2200 and 2600 RPMs.")]
sint16    RPM_2200_2600_Pct;

[Description("Percentage of time the fan speed was between 2600 and 2900 RPMs.")]
sint16    RPM_2600_2900_Pct;

[Description("Percentage of time the fan speed was between 2900 and 3100 RPMs.")]
sint16    RPM_2900_3100_Pct;

[Description("Percentage of time the fan speed was between 3100 and 3300 RPMs.")]
sint16    RPM_3100_3300_Pct;

[Description("Percentage of time the fan speed was between 3300 and 3600 RPMs.")]
sint16    RPM_3300_3600_Pct;

[Description("Percentage of time the fan speed was between 3600 and 3900 RPMs.")]
sint16    RPM_3600_3900_Pct;

[Description("Percentage of time the fan speed was between 3900 and 4200 RPMs.")]
sint16    RPM_3900_4200_Pct;

[Description("Percentage of time the fan speed was between 4200 and 4600 RPMs.")]
sint16    RPM_4200_4600_Pct;

[Description("Percentage of time the fan speed was between 4600 and 5100 RPMs.")]
sint16    RPM_4600_5100_Pct;

[Description("Percentage of time the fan speed was between 5100 and 5600 RPMs.")]
sint16    RPM_5100_5600_Pct;

[Description("Percentage of time the fan speed was between 5600 and 6200 RPMs.")]
sint16    RPM_5600_6200_Pct;

[Description("Percentage of time the fan speed was between 6200 and 7000 RPMs.")]
sint16    RPM_6200_7000_Pct;
```
```cpp
class DPO_HardwareInfoToNBFan {
    [key] DPO_HardwareInfo REF Antecedent;
    [key] DPO_NBFan REF Dependent;
};

class DPO_DTFan {
    [Description("Desktop fan speed statistics.")] string HardwareInfoGUID;

    [Description("Desktop fan index number starting from 0.")] sint16 Index;
    [Description("Location where the fan is present in the system.")] string Location;
    [Description("Percentage of time fan rpm was non-zero.")] sint16 Fan_Duty_Cycle_Pct;
    [Description("Fan speed when the summary log was generated.")] sint32 RPM;
    [Description("Peak fan speed.")] sint32 Peak_Fan_RPM;
    [Description("Average fan speed.")] real32 Average_Fan_RPM;

    [Description("Percentage of time the fan speed was between 0 and 500 RPMs.")] sint16 RPM_0_500_Pct;
    [Description("Percentage of time the fan speed was between 500 and 900 RPMs.")] sint16 RPM_500_900_Pct;
    [Description("Percentage of time the fan speed was between 900 and 1100 RPMs.")] sint16 RPM_900_1100_Pct;
    [Description("Percentage of time the fan speed was between 1100 and 1300 RPMs.")] sint16 RPM_1100_1300_Pct;
    [Description("Percentage of time the fan speed was between 1300 and 1600 RPMs.")] sint16 RPM_1300_1600_Pct;
    [Description("Percentage of time the fan speed was between 1600 and 1900 RPMs.")] sint16 RPM_1600_1900_Pct;
    [Description("Percentage of time the fan speed was between 1900 and 2300 RPMs.")] sint16 RPM_1900_2300_Pct;
    [Description("Percentage of time the fan speed was between 2300 and 2700 RPMs.")] sint16 RPM_2300_2700_Pct;
    [Description("Percentage of time the fan speed was between 2700 and 3100 RPMs.")] sint16 RPM_2700_3100_Pct;
};
```
sint16    RPM_3100_3500_Pct;

sint16    RPM_3500_4000_Pct;

sint16    RPM_4000_4500_Pct;

sint16    RPM_4500_5000_Pct;

sint16    RPM_5000_5500_Pct;

sint16    RPM_5500_6000_Pct;

sint16    RPM_GT6000_Pct;

/***********************************************************/
*    DPO_HardwareInfoToDTFan
*    This class associates DPO_DTFan instance(s) with an
*    instance of DPO_HardwareInfo.
**************************************************************/
class DPO_HardwareInfoToDTFan
{
    [key] DPO_HardwareInfo REF    Antecedent;
    [key] DPO_DTFan    REF    Dependent;
};

/***********************************************************/
*    DPO_Thermistor
*    This has the thermal information from a summary log. There
*    may be multiple instances of this class for each summary file.
**************************************************************/
class DPO_Thermistor
{
    [Description("Unique ID from the summary file.", Key,
    Description("Thermistor index number starting from 0.", Key,
    Description("Thermistor location eg CPU, Memory etc."))]
    string    HardwareInfoGUID;

    sint16    Index;

    string    Location;

    [Description("Temperature read from the thermistor when the summary log was generated.",
    Description("Maximum temperature read from the thermistor.",
    Description("Average temperature read from the thermistor.",
    Description("Minimum temperature read from the thermistor.",
    Description("Standard deviation of temperature read from the thermistor.")
}

APPENDIX B - WMI Class Definition Files
sint16    Temp_0_30C_Pct;

sint16    Temp_30_40C_Pct;

sint16    Temp_40_50C_Pct;

sint16    Temp_50_60C_Pct;

sint16    Temp_60_70C_Pct;

sint16    Temp_70_80C_Pct;

sint16    Temp_80_90C_Pct;

sint16    Temp_90_100C_Pct;

sint16    Temp_GT100C_Pct;

/**
 * DPO_HardwareInfoToThermistor
 * This class associates DPO,Thermistor instance(s) with an
 * instance of DPO_HardwareInfo.
 */

class DPO_HardwareInfoToThermistor
{
    [key] DPO_HardwareInfo REF    Antecedent;
    [key] DPO_Thermistor    REF    Dependent;
}

/**
 * DPO_Logical_Processor
 * This has the logical processor information from a summary log.
 * There may be multiple instances of this class for each summary
 * file.
 */

class DPO_Logical_Processor
{
    [Description("Logical processors statistics.")],
    Dynamic,Provider("DPOProv") ]

    string    HardwareInfoGUID;

    [Description("Unique ID from the summary file.")],
    Key ]

    sint16    Index;

    [Description("Percentage of time the logical processor was used, ie. when the CPU consumption was non-zero.")]
    sint16    Used_Pct;

    [Description("Average processor utilization.")]
    sint16    Avg_Utilization_Pct;
}
/**
 * DPO_HardwareInfoToLogical_Processor
 * This class associates DPO_Logical_Processor instance(s) with an instance of DPO_HardwareInfo.
 */

[Association : ToInstance,
Description("This class associates DPO_Logical_Processor instance(s) with an instance of DPO_HardwareInfo"),
dynamic:ToInstance,
PROVIDER("DPOProv"):ToInstance]
class DPO_HardwareInfoToLogical_Processor
{
[key] DPO_HardwareInfo    REF    Antecedent;
[key] DPO_Logical_Processor    REF    Dependent;
};

/**
 * DPO_Disk
 * This has the physical disk information from a summary log. There may be multiple instances of this class for each summary file.
 */

[Description("Information for each physical disk found on the system."), Dynamic, Provider("DPOProv")]
class DPO_Disk
{
    [Description("Unique ID from the summary file.")],
    Key
    string    HardwareInfoGUID;

    [Description("Index of the physical disk starting from 0.")],
    Key
    sint16    Index;

    [Description("Name of the disk.")]
    string    Name;

    [Description("Disk model number.")]
    string    Make_Model;

    [Description("Total disk size in MBs.")]
    sint32    Size_MB;

    [Description("Disk ePPID.")]
    string    ePPID;

    [Description("Unique ID assigned to this disk instance.")]
    string    DiskGUID;

    [Description("Percentage of time the disk was busy in read operations.")]
    sint16    Read_Time_Pct;

    [Description("Percentage of time the disk was busy in write operations.")]
    sint16    Write_Time_Pct;

    [Description("Percentage of time the disk was idle.")]
    sint16    Idle_Time_Pct;

    [Description("Total data read from the disk in MB.")]
    sint32    Bytes_Read_MB;

    [Description("Total data written to the disk in MB.")]
    sint32    Bytes_Write_MB;
};
[Association : ToInstance,
Description("This class associates DPO_Disk instance(s) with " " an instance of
DPO_HardwareInfo"),
dynamic:ToInstance,
PROVIDER("DPOProv") : ToInstance
]
class DPO_HardwareInfoToDisk
{
[key] DPO_HardwareInfo REF Antecedent;
[key] DPO_Disk REF Dependent;
};

/**************************************************************
*    DPO_Partition
*    This has the logical partition information from a summary log.
*    There may be multiple instances of this class for each summary
*    file.
**************************************************************/
[Description("Information for each partition found on a disk.")],
Dynamic,Provider("DPOProv")
] class DPO_Partition
{
[Description("Unique ID from the summary file.")],
Key
} string HardwareInfoGUID;

[Description("Unique ID assigned to the physical disk instance to which this partition
belongs.")],
Key
} string DiskGUID;

[Description("Partition index number starting from 0.")],
Key
}sint16 Index;

[Description("Partition name, eg C:.")]
string Name;

[Description("Total size of the partition in MBs.")]
sint32 Size_MB;
};

/**************************************************************
*    DPO_DiskToPartition
*    This class associates DPO_Partition instance(s) with an
*    instance of DPO_Disk.
**************************************************************/
[Association : ToInstance,
Description("This class associates DPO_Partition instance(s) " " with an instance of
DPO_Disk"),
dynamic:ToInstance,
PROVIDER("DPOProv") : ToInstance
] class DPO_DiskToPartition
{
[key] DPO_Disk REF Antecedent;
[key] DPO_Partition REF Dependent;
};

/**************************************************************
*    DPO_LanAdapter
*    This has the lan adapter information from a summary log. There
*    may be multiple instances of this class for each summary file.
***************************************************************/
[Description("LAN adapter information and statistics.")],
Dynamic,Provider("DPOProv")
]
class DPO_LanAdapter
{
    [Description("Unique ID from the summary file.")],
    Key
    string    HardwareInfoGUID;

    [Description("LAN adapter index number starting from 0.")],
    Key
    sint16    Index;

    [Description("LAN adapter name.")]
    string    Name;

    [Description("LAN adapter's MAC address.")]
    string    MAC;

    [Description("Percentage of time the adapter was busy when the system was on AC.")]
    sint16    ActivityAC_Pct;

    [Description("Percentage of time the adapter was busy when the system was on battery.")]
    sint16    ActivityDC_Pct;
};

class DPO_HardwareInfoToLanAdapter
{
    [Association : ToInstance, Description("This class associates DPO_LanAdapter instance(s) with an instance of DPO_HardwareInfo.")]
    dynamic:ToInstance,
    PROVIDER("DPOProv"):ToInstance
}

class DPO_WlanAdapter
{
    [Description("Unique ID from the summary file.")],
    Key
    string    HardwareInfoGUID;

    [Description("Wireless LAN adapter index number starting from 0.")],
    Key
    sint16    Index;

    [Description("Wireless LAN adapter name.")]
    string    Name;

    [Description("Wireless LAN adapter's MAC address.")]
    string    MAC;

    [Description("Percentage of time the radio was off when the system was on AC.")]
    sint16    WlanRadioOffAC_Pct;
};

APPENDIX B - WMI Class Definition Files
[Description("Percentage of time the WLAN adapter was connected when the system was on AC.")]
sint16 WlanConnectedAC_Pct;

[Description("Percentage of time the adapter was not connected when the system was on AC.")]
sint16 WlanDisconnectedAC_Pct;

[Description("Percentage of time the radio was off when the system was on battery.")]
sint16 WlanRadioOffDC_Pct;

[Description("Percentage of time the WLAN adapter was connected when the system was on battery.")]
sint16 WlanConnectedDC_Pct;

[Description("Percentage of time the adapter was not connected when the system was on battery.")]
sint16 WlanDisconnectedDC_Pct;

/
**************************************************
*    DPO_HardwareInfoToWlanAdapter
*    This class associates DPO_WlanAdapter instance(s) with an
*    instance of DPO_HardwareInfo.
**************************************************/

[Association : ToInstance, 
  Description("This class associates DPO_WlanAdapter instance(s) " " with an instance of
  DPO_HardwareInfo"),
  dynamic:ToInstance, 
  PROVIDER("DPOProv")]:ToInstance
]
class DPO_HardwareInfoToWlanAdapter
{
  [key] DPO_HardwareInfo REF Antecedent;
  [key] DPO_WlanAdapter REF Dependent;
};

/
**************************************************
*    DPO_Smart
*    This has the SMART information from a summary log. There
*    may be multiple instances of this class for each summary file.
**************************************************/

[Description("SMART data from all disks (if reported by the disk)."),
  Dynamic,Provider("DPOProv") ]
class DPO_Smart
{
  [ 
    Description("Unique ID from the summary file."),
    Key
  ]
  string HardwareInfoGUID;

  [ 
    Description("Smart data index number starting from 0."),
    Key
  ]
sint16 Index;

  [ 
    Description("Name eg, SMART0."))
  ]
  string Name;

  [ 
    Description("Disk Model number."))
  ]
  string Model;

  [ 
    Description("Average disk temperature read using SMART."))
  ]
  real32 Temp_Avg;

  [ 
    Description("Standard deviation of disk temperature read using SMART."))
  ]
  real32 Temp_Std_Dev;

  [ 
    Description("Minimum disk temperature read using SMART."))
  ]
sint16 Temp_Min;
```
 sint16    Temp_Max;
 sint16    Temp_0_30_Pct;
 sint16    Temp_30_40_Pct;
 sint16    Temp_40_50_Pct;
 sint16    Temp_50_60_Pct;
 sint16    Temp_60_70_Pct;
 sint16    Temp_70_80_Pct;
 sint16    Temp_GT_80_Pct;
 sint32    Shock_Events;
 uint8    Shock_Events_Normalized;
 uint8    Shock_Events_Worst;
 uint8    Shock_Events_Threshold;
 sint64    Blks_Read;
 uint8    Blks_Read_Normalized;
 uint8    Blks_Read_Worst;
 uint8    Blks_Read_Threshold;
 sint64    Load_Unload_Cycle_Count;
 uint8    Load_Unload_Cycle_Count_Normalized;
 uint8    Load_Unload_Cycle_Count_Worst;
 uint8    Load_Unload_Cycle_Count_Threshold;
```
uint8     Load_Unload_Cycle_Count_Normalized;
uint8     Load_Unload_Cycle_Count_Worst;
uint8     Load_Unload_Cycle_Count_Threshold;
sint64    Power_On_Hours;
uint8     Power_On_Hours_Normalized;
uint8     Power_On_Hours_Worst;
uint8     Power_On_Hours_Threshold;
sint64    ReAlloc_Sector_Count;
uint8     ReAlloc_Sector_Count_Normalized;
uint8     ReAlloc_Sector_Count_Worst;
uint8     ReAlloc_Sector_Count_Threshold;
sint64    Head_Flying_Hours;
uint8     Head_Flying_Hours_Normalized;
uint8     Head_Flying_Hours_Worst;
uint8     Head_Flying_Hours_Threshold;
sint64    Raw_Read_Error_Rate;
uint8     Raw_Read_Error_Rate_Normalized;
uint8     Raw_Read_Error_Rate_Worst;
uint8     Raw_Read_Error_Rate_Threshold;
sint64    Spin_Up_Time;
uint8     Spin_Up_Time_Normalized;
uint8     Spin_Up_Time_Worst;
uint8     Spin_Up_Time_Threshold;
sint64    Free_Fall_Count;
sint64    Free_Fall_Count;
uint8    Free_Fall_Count_Normalized;
[Description("Free fall count (worst value).")]]
uint8    Free_Fall_Count_Worst;
[Description("Free fall count (threshold value).")]
uint8    Free_Fall_Count_Threshold;
[Description("Power cycle count.")] sint64    Power_Cycle_Count;
[Description("Power cycle count (normalized value).")]
uint8    Power_Cycle_Count_Normalized;
[Description("Power cycle count (worst value).")]
uint8    Power_Cycle_Count_Worst;
[Description("Power cycle count (threshold value).")]
uint8    Power_Cycle_Count_Threshold;
[Description("Program fail count.")] sint64    Program_Fail_Count;
[Description("Program fail count (normalized value).")]
uint8    Program_Fail_Count_Normalized;
[Description("Program fail count (worst value).")]
uint8    Program_Fail_Count_Worst;
[Description("Program fail count (threshold value).")]
uint8    Program_Fail_Count_Threshold;
[Description("Erase fail count.")] sint64    Erase_Fail_Count;
[Description("Erase fail count (normalized value).")]
uint8    Erase_Fail_Count_Normalized;
[Description("Erase fail count (worst value).")]
uint8    Erase_Fail_Count_Worst;
[Description("Erase fail count (threshold value).")]
uint8    Erase_Fail_Count_Threshold;
[Description("Wear leveling count.")] sint64    Wear_Leveling_Count;
[Description("Wear leveling count (normalized value).")]
uint8    Wear_Leveling_Count_Normalized;
[Description("Wear leveling count (worst value).")]
uint8    Wear_Leveling_Count_Worst;
[Description("Wear leveling count (threshold value).")]
uint8    Wear_Leveling_Count_Threshold;
[Description("User reserved block count.")] sint64    User_Rsvd_Block_Count;
[Description("User reserved block count (normalized value).")]
uint8    User_Rsvd_Block_Count_Normalized;
[Description("User reserved block count (worst value).")]
uint8    User_Rsvd_Block_Count_Worst;
[Description("User reserved block count (threshold value).")]
uint8    User_Rsvd_Block_Count_Threshold;
[Description("User reserved block count (SSD Total).")]
uint8    User_Rsvd_Block_Count_Total;
[Description("User reserved block count (SSD Total) (normalized value).")]}
import {DPO_HardwareInfo} from 'dpo-hardwareinfo';

export class DPO_HardwareInfoToSmart {
  constructor(dpoSmart) {
    // Create a new instance of DPO_Smart
    this.dpoSmart = dpoSmart;
  }

  // Method to associate DPO_Smart instance(s) with an instance of DPO_HardwareInfo
  associatemethod(hardwareInfo) {
    // Associate the DPO_Smart instance(s) with the hardware info
    // Implementation details...
  }

  // Method to get the instance of DPO_HardwareInfo
  getHardwareInfo() {
    // Return the instance of DPO_HardwareInfo
    return this.dpoHardwareInfo;
  }

  // Method to get the instance of DPO_Smart
  getDPOSmart() {
    // Return the instance of DPO_Smart
    return this.dpoSmart;
  }
}

// Example usage
const dpoSmart = new DPO_Smart();
const dpoHardwareInfo = new DPO_HardwareInfo();
const dpoHardwareInfoToSmart = new DPO_HardwareInfoToSmart(dpoSmart);

// Associate DPO_Smart instance(s) with DPO_HardwareInfo instance
dpoHardwareInfoToSmart.associatemethod(dpoHardwareInfo);

// Get the instance of DPO_Smart
const instanceOfDPOSmart = dpoHardwareInfoToSmart.getDPOSmart();

// Get the instance of DPO_HardwareInfo
const instanceOfDPOHardwareInfo = dpoHardwareInfoToSmart.getHardwareInfo();
class DPO_HardwareInfoToSmart
{
    [key] DPO_HardwareInfo REF Antecedent;
    [key] DPO_Smart REF Dependent;
};

class DPO_HardwareInfoToDIMM
{
    [key] DPO_HardwareInfo REF Antecedent;
    [key] DPO_DIMM REF Dependent;
};

class DPO_DIMM
{
    [Description("Unique ID from the summary file.")]
    Key
    string HardwareInfoGUID;

    [Description("DIMM index number starting from 0.")]
    Key
    sint16 Index;

    [Description("DIMM name.")]
    string Name;

    [Description("DIMM manufacturer's name.")]
    string Manufacturer;

    [Description("DIMM part number.")]
    string Part;

    [Description("DIMM location.")]
    string Location;

    [Description("DIMM serial number.")]
    string Serial;
};

class DPO_HardwareInfoToDIMM
{
    [Association : ToInstance,
    Description("This class associates DPO_DIMM instance(s) with an instance of DPO_HardwareInfo"),
    dynamic:ToInstance,
    PROVIDER("DPOProv"):ToInstance]
    class DPO_HardwareInfoToDIMM
    {
        [key] DPO_HardwareInfo REF Antecedent;
        [key] DPO_DIMM REF Dependent;
    };

APPENDIX B - WMI Class Definition Files
class DPO_Logical_Drive_Info_New
{
    string HardwareInfoGUID;
    sint16 Index;
    string Name;
    sint64 Size_MB;
    sint64 Freespace_MB;
};

class DPO_HardwareInfoToLogicalDriveInfoNew
{
    DPO_HardwareInfo REF Antecedent;
    DPO_Logical_Drive_Info_New REF Dependent;
};

class DPO_CrashInfo
{
    string HardwareInfoGUID;
    sint16 Index;
    string BugCheck_Time;
    string BugCheck_String;
};
[Description("MiniDump File Name.")]
string Minidump_FileName;

[Description("MiniDump File Data Length")]
uint32 Minidump_DataLen;

[Description("MiniDump File Binary Data")]
uint8 Minidump_Data[];

[Description("Bug check stack frame 1")]
string BugCheck(Stack1);

[Description("Bug check stack frame 2")]
string BugCheck(Stack2);

[Description("Bug check stack frame 3")]
string BugCheck(Stack3);

[Description("Bug check stack frame 4")]
string BugCheck(Stack4);

[Description("Bug check stack frame 5")]
string BugCheck(Stack5);

};

/**************************************************************/
/* DPO_HardwareInfoToCrashInfo                               */
/* This class associates DPO_CrashInfo instance(s) with an   */
/* instance of DPO_HardwareInfo.                             */
/**************************************************************/
[Association : ToInstance,
 Description("This class associates DPO_CrashInfo instance(s) with an instance of DPO_HardwareInfo"),
 dynamic:ToInstance,
 PROVIDER("DPOProv") : ToInstance]
class DPO_HardwareInfoToCrashInfo
{
[key] DPO_HardwareInfo REF Antecedent;
[key] DPO_CrashInfo REF Dependent;
};

/**************************************************************/
/* DPO_FreeFall                                              */
/* This has Free fall information from a summary log. Right  */
/* now, there is only one instance of this class for each summary file but that may change in the future. */
/**************************************************************/
[Description("Free fall information from the summary log file.")],
 Dynamic,Provider("DPOProv")
class DPO_FreeFall
{
[Description("Unique ID from the summary file.")],
 Key
} string HardwareInfoGUID;
[Description("Number of times free fall condition was detected since last summary file was generated.")]
sint16 FreeFallCount;

/**************************************************************/
/* DPO_HardwareInfoToFreeFall                               */
/* This class associates DPO_FreeFall instance(s) with an     */
/* instance of DPO_HardwareInfo.                             */
/**************************************************************/
[Association : ToInstance,
 [APPENDIX B - WMI Class Definition Files 53]
Description("This class associates DPO_FreeFall instance(s) with an instance of DPO_HardwareInfo"),
{  
  [key] DPO_HardwareInfo REF Antecedent;
  [key] DPO_FreeFall REF Dependent;
};

/*******************************************************************************
* DPO_Cable
* This has the cable log information from a summary log. There may be multiple instances of this class for each summary file.
*******************************************************************************/
[Description("Cable logs: List of all cables attached, required but not connected in the system."), Dynamic,Provider("DPOProv") ]
class DPO_Cable
  {
  [Description("Unique ID from the summary file.")],
  Key
  string HardwareInfoGUID;

  [Description("Cable index number, starting from 0.")],
  Key
  sint16 Index;

  [Description("Name of cable.")]
  string Name;

  [Description("Cable's connection status.")]
  string Status;
};

/*******************************************************************************
* DPO_HardwareInfoToCableLogs
* This class associates DPO_HardwareInfoToCable instance(s) with an instance of DPO_HardwareInfo.
*******************************************************************************/
[Association : ToInstance,
  Description("This class associates DPO_HardwareInfoToCable instance(s) with an instance of DPO_HardwareInfo."),
  dynamic:ToInstance,
  PROVIDER("DPOProv") : ToInstance]
class DPO_HardwareInfoToCable
{
  [key] DPO_HardwareInfo REF Antecedent;
  [key] DPO_Cable REF Dependent;
};

/*******************************************************************************
* DPO_CableChangeHistory
* This has the cable change history information from a summary log. There may be multiple instances of this class for each summary file.
*******************************************************************************/
[Description("Information for status change for a cable."), Dynamic,Provider("DPOProv") ]
class DPO_CableChangeHistory
{
  [Description("Name of cable.")]
  string Name;

  [Description("Timestamp when the change in cable status was noted.")]
  string Timestamp;

  [Description("Cable's connection status.")]
  string Status;
### DPO_CableToCableChangeHistory

This class associates DPO_CableChangeHistory instance(s) with an instance of DPO_Cable.

```plaintext
[Association : ToInstance,
  Description("This class associates DPO_CableChangeHistory instance(s) with an instance of DPO_Cable"),
  dynamic:ToInstance,
  PROVIDER("DPOProv") : ToInstance]
```

class DPO_CableToCableChangeHistory
{
  [key] DPO_Cable    REF    Antecedent;
  [key] DPO_CableChangeHistory    REF    Dependent;
};
```

### DPO_BTModule

This has the bluetooth module information from a summary log. There may be multiple instances of this class for each summary file.

```plaintext
[Description("Bluetooth module information and statistics.")],
  Dynamic, Provider("DPOProv")]
```

class DPO_BTModule
{
  [Key]
  Description("Unique ID from the summary file.")]
  string    HardwareInfoGUID;

  [Key]
  Description("Bluetooth module index number starting from 0.")]
  sint16    Index;

  [Description("Bluetooth module name.")]
  string    Name;

  [Description("Bluetooth module's address.")]
  string    Address;

  [Description("Percentage of time the radio was on when the system was on AC.")]
  sint16    BTRadioOnAC_Pct;

  [Description("Percentage of time the bluetooth module was connected when the system was on AC.")]
  sint16    BTConnectedAC_Pct;

  [Description("Percentage of time the module was not connected when the system was on AC.")]
  sint16    BTDisconnectedAC_Pct;

  [Description("Percentage of time the radio was on when the system was on battery.")]
  sint16    BTRadioOnDC_Pct;

  [Description("Percentage of time the bluetooth module was connected when the system was on battery.")]
  sint16    BTConnectedDC_Pct;

  [Description("Percentage of time the module was not connected when the system was on battery.")]
  sint16    BTDisconnectedDC_Pct;
};
```

### DPO_HardwareInfoToBTModule

This class associates DPO_BTModule instance(s) with an instance of DPO_HardwareInfo.

```plaintext
[Association : ToInstance,
  Description("This class associates DPO_BTModule instance(s) with an instance of DPO_HardwareInfo.")]
```
[Association : ToInstance,
Description("This class associates DPO_BTModule instance(s) with an instance of DPO_HardwareInfo"),
dynamic:ToInstance,
PROVIDER("DPOProv"):ToInstance
]
class DPO_HardwareInfoToBTModule
{
[key] DPO_HardwareInfo REF Antecedent;
[key] DPO_BTModule REF Dependent;
}

/**************************************************************/
*    DPO_IntelPerf
*    This has the Intel performance information from a summary log. There
*    may be multiple instances of this class for each summary file.
**************************************************************/
[Description("Intel performance information and statistics.")],
Dynamic,Provider("DPOProv")
]
class DPO_IntelPerf
{
[
Description("Unique ID from the summary file."),
Key
] string HardwareInfoGUID;
[
Description("Processor number starting from 0."),
Key
] sint16 Index;
[Description("Minimum active relative frequency of the processor.")]
real32 Min_ActiveRelativeFreq;
[Description("Maximum active relative frequency of the processor.")]
real32 Max_ActiveRelativeFreq;
[Description("Average active relative frequency of the processor.")]
real32 Avg_ActiveRelativeFreq;
[Description("Percentage of time the processor was in turbo mode when the system was on AC.")]
real32 TurboResidencyACPct;
[Description("Percentage of time the processor was in turbo mode when the system was on battery.")]
real32 TurboResidencyDCPct;
};

/**************************************************************/
*    DPO_HardwareInfoToIntelPerf
*    This class associates DPO_IntelPerf instance(s) with an
*    instance of DPO_HardwareInfo.
**************************************************************/
[Association : ToInstance,
Description("This class associates DPO_IntelPerf instance(s) with an instance of DPO_HardwareInfo.

**************************************************************/
class DPO_HardwareInfoToIntelPerf
{
[key] DPO_HardwareInfo REF Antecedent;
[key] DPO_IntelPerf REF Dependent;
};

/**************************************************************/
*    DPO_Graphics
*    This has the graphics information from a summary log. There
*    may be multiple instances of this class for each summary file.
class DPO_Graphics
{
    string    HardwareInfoGUID;

    sint16    Index;

    sint16    Min_GpuUtilization;
    sint16    Max_GpuUtilization;
    real32    Avg_GpuUtilization;
    real32    GpuUtilization_0_Pct;

    sint16    Min_MemUtilization;
    sint16    Max_MemUtilization;
    real32    Avg_MemUtilization;
    real32    MemUtilization_0_Pct;

    sint16    Min_EngineUtilization;
    sint16    Max_EngineUtilization;
    real32    Avg_EngineUtilization;
    real32    EngineUtilization_0_Pct;

    sint16    Min_BusUtilization;
    sint16    Max_BusUtilization;
    real32    Avg_BusUtilization;
    real32    BusUtilization_0_Pct;

    sint16    Min_FanSpeedPct;
    sint16    Max_FanSpeedPct;
    real32    Avg_FanSpeedPct;
real32 FanSpeedPct_0_Pct;

sint16 Min_Temperature;

sint16 Max_Temperature;

real32 Avg_Temperature;

class DPO_HardwareInfoToGraphics
{
    [key] DPO_HardwareInfo REF Antecedent;
    [key] DPO_Graphics REF Dependent;
};

class DPO_Info
{
    string ProductVersion;

    string LastCheckForUpdateTime;

    string LastSystemUpdateTime;

};
string LastCheckForProfiles;
};

/******************************************************************************************************************
*DPO_TriggeredProfiles
******************************************************************************************************************/
Description("DPO Profiles that have triggered"),
Dynamic, Provider("DPOProv")
class DPO_TriggeredProfiles
{
    [Description("Unique ID of profile"),
    Key]
    string ProfileGUID;

    [Description("Name of profile")]
    string ProfileName;

    [Description("Unique ID of policy that triggered"),
    Key]
    string PolicyGUID;

    [Description("Name of policy that triggered")]
    string PolicyName;

    [Description("Date/Time of trigger"), key]
    string TriggeredAt;
};

@Autowired

/******************************************************************************************************************
*DPO_Profiles
******************************************************************************************************************/
[Description("DPO Profiles"),
Dynamic, Provider("DPOProv")
class DPO_Profiles
{
    [Description("Unique ID"),
    Key]
    string ProfileGUID;

    [Description("Name")]
    string ProfileName;

    [Description("Active")]
    string Active;
};

@Autowired

/******************************************************************************************************************
*DPO_SmartAlerts
******************************************************************************************************************/
[Description("DPO Smart Alerts"),
Dynamic, Provider("DPOProv")
class DPO_SmartAlerts
{
    [Description("Policy that triggered")]
    string PolicyGUID;

    [Description("Name")]
    string PolicyName;

    [Description("Date/Time of trigger")]
    string TriggeredAt;
};
instance of Win32Provider as $P
{
Clsid = "{C4ABD5F1-1260-4192-BF0B-11909C172043}";

Name = "DPOProv";
HostingModel = "NetworkServiceHost";
};

instance of InstanceProviderRegistration
{
Provider = $P;
SupportsGet = TRUE;
SupportsPut = FALSE;
SupportsDelete = FALSE;
SupportsEnumeration = TRUE;

// we want WMI to do query parsing QuerySupportLevels = NULL;
};

instance of MethodProviderRegistration
{
Provider = $P;
}