



Integrating Dell Command | Monitor with Microsoft® System Center Configuration Manager 2012

Dell Engineering
October 2014

Revisions

Date	Description
Oct' 2014	Initial release

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES.

THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2014 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

PRODUCT WARRANTIES APPLICABLE TO THE DELL PRODUCTS DESCRIBED IN THIS DOCUMENT MAY BE FOUND AT:

<http://www.dell.com/learn/us/en/19/terms-of-sale-commercial-and-public-sector> Performance of network reference architectures discussed in this document may vary with differing deployment conditions, network loads, and the like. Third party products may be included in reference architectures for the convenience of the reader. Inclusion of such third party products does not necessarily constitute Dell's recommendation of those products Please consult your Dell representative for additional information. Trademarks used in this text:

Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManag™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Other Dell trademarks may be used in this document. Cisco Nexus®, Cisco MDS®, Cisco NX-OS®, and other Cisco Catalyst® are registered trademarks of Cisco System Inc. EMC VNX®, and EMC Unisphere® are registered trademarks of EMC Corporation. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter® and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation. Broadcom® and NetXtreme® are registered trademarks of Broadcom Corporation. Qlogic is a registered trademark of QLogic Corporation. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and/or names or their products and are the property of their respective owners. Dell disclaims proprietary interest in the marks and names of others.



Table of contents

- Revisions..... 2
- Acknowledgements..... 4
- Feedback 4
- Executive summary 4
- 1 Introduction..... 5
 - 1.1 Purpose and scope..... 5
 - 1.2 Terminology..... 6
- 2 Using DC | M in ConfigMgr 7
 - 2.1 Namespace Information..... 7
 - 2.2 Accessing DC | M information in ConfigMgr 7
 - 2.2.1 Inventory using DC | M MOF..... 7
 - 2.2.2 Queries Compiled in SCCM 12
- A Appendix A 25
 - A.1.1 OMCI_SMS_DEF.mof 25
- B Additional resources..... 65



Acknowledgements

This white paper was produced by the following members of the Dell Client Enterprise team:

Engineering: Sharmad Naik

Test/Quality: Karthik Chandran & Palani Raja

Enhancement: Mainak Roy

Feedback

We encourage readers of this publication to provide feedback on the quality and usefulness of this by logging information on [OMCI TechCenter forum](#).

Executive summary

System Center Configuration Manager (ConfigMgr) is a Microsoft® systems management software product for managing large groups of computers running on different Operating Systems. Configuration Manager provides remote control, patch management, software distribution, operating system deployment, network access protection and hardware and software inventory related information.

Dell Command | Monitor (DC | M) is a software that allows systems management application programs to access information about the client computer, monitor the status of the client computer, or change the state of the client computer, such as shutting it down remotely. DC | M uses the Common Information Model (CIM) and Web Based Enterprise Management (WBEM). These are industry standard systems management technologies defined by the Distributed Management Task Force (DMTF). Microsoft® Windows® Management Instrumentation (WMI) is Microsoft's implementation of CIM instrumentation. DC | M provides data to WMI, which is the common interface for the WMI management applications.

Although DC | M has numerous features, its primary purpose is to package and provide access to the information requested by WMI, which in turn provides the information to systems management application programs such as Microsoft System Center Configuration Manager (ConfigMgr).

The CIM schema is defined by a Managed Object Format (MOF) file, which provides a standardized model for describing management information between clients in a management system environment. A MOF file is not bound to any particular implementation, rather it allows information interchange between multiple management systems and clients.



1 Introduction

Client Instrumentation refers to software applications that enable remote management of a client system. The Dell Command | Monitor (Command | Monitor) software enables remote management using application programs to access the Enterprise Client system information, monitor the status, or change the state of the system such as remotely shutting down the system. Command | Monitor uses key system parameters through standard interfaces allowing administrators to manage inventory, monitor system health, and gather information about deploying Enterprise client systems.

Command | Monitor manages client systems using the Common Information Model (CIM) standard and SNMP, which are management protocols. This reduces the total cost of ownership, improves security, and provides a holistic approach to manage all the devices, including clients, servers, storage, network, and software devices.

Using CIM you can access Command | Monitor through Web Services for Management Standards (WSMAN).

Command | Monitor contains the underlying driver set that collects client system information from different sources, including the BIOS, CMOS, System Management BIOS (SMBIOS), System Management Interface (SMI), operating system, Application programming interface (APIs), Dynamic-link library (DLLs), and registry settings. Command | Monitor fetches this information through the CIM Object Manager (CIMOM) interface, Windows Management Instrumentation (WMI) stacks or SNMP agent.

Command | Monitor enables IT administrators to remotely collect asset information, modify CMOS settings, receive proactive notifications about potential fault conditions, and get alerts for potential security breaches. These alerts are available as events in the event log, CIM Indication or received as SNMP traps after importing the MIB file and monitoring it.

Command | Monitor is used to gather asset inventory from the system, including BIOS settings, through CIM implementation or SNMP agent. It can be integrated into a console such as Microsoft® System Center Configuration Manager by directly accessing the CIM information, or through other console vendors which have implemented the Command Monitor integration.

Apart of that, you can create custom scripts for other important aspects. These scripts can be used to monitor inventory, BIOS settings, and system health.

1.1 Purpose and scope

Although Command | Monitor has numerous features, its primary purpose is to package and provide access to information requested by WMI, which in turn provides the information to systems management application programs such as the Microsoft® System Center Configuration Manager (ConfigMgr) tools.

This whitepaper provides details on how an IT administrator can exploit the information provided by Command | Monitor within ConfigMgr to provide enhanced reporting for managing Dell business client platforms. This document assumes the reader is familiar with the DCM / OMCI usage, Microsoft SCCM and



system administration. The scope of this paper is restricted to inventory and doesn't include deployment of Dell Command | Monitor on Client Systems using ConfigMgr.

1.2 Terminology

The following terms will be used throughout this document

SCCM System Center Configuration Manager

SSRS SQL Server Reporting Service



2 Using DC | M in ConfigMgr

2.1 Namespace Information

The namespace for access to the Dell OMCI classes is "root\dcim\sysman".

2.2 Accessing DC | M information in ConfigMgr

There are two ways that DC | M data may be made available in ConfigMgr. Section 2.2.1 shows how importing the DC | M mof file OMCI_SMS_DEF.mof can help display inventory information of Dell Enterprise Client System. Section 2.2.2 and Section 2.2.3 extend the integration by providing a number of pre-defined ConfigMgr web reports and queries, as well as explaining how these may be exploited in the ConfigMgr console.

Section 2.2.2 contains a set of sample ConfigMgr web reports that correspond to the sample MOF in Section 2.2.1. These reports are based on classes and properties enabled in the OMCI SMS MOF. Section 2.2.3 explains how the sample queries can be compiled into the ConfigMgr site server using mofcomp and the MOF formatted sample queries or pasted into a new web report in the ConfigMgr console.

Section 2.2.3 contains example ConfigMgr queries based on the OMCI SMS MOF in Section 2.2.1.

2.2.1 Inventory using DC | M MOF

After the installation of DC | M is completed, If the default install path is chosen than the inventory of is located in the following folder %ProgramFiles%\Dell\Command Monitor\ssa\omacim. Following are the steps to import the file into SCCM 2012:

1. Copy the OMCI SCCM mof file on the server.
2. Modify the MOF to change any "**FALSE**" to "**TRUE**" (or vice versa) for the desired reporting within ConfigMgr. This step allows customization of reports to the level of detail required. NOTE: In order for a property to be collected for reporting purposes, the value of "**SMS_Report**" for that property must be set to "**TRUE**"; additionally, the value of "**SMS_Report**" for the class that contains the property must also be set to "**TRUE**". Save the mof.
3. Open the SCCM console.
4. Navigate to the Administration Feature Tab/Menu and follow the below step/order to import the Mof into SCCM.
 - a. Click on the Administration Tab
 - b. Click on Client Setting
 - c. Default client settings (On Right side pane)
 - d. Right click and select Properties
 - e. Hardware inventory menu
 - f. Click on Set classes (On Right side pane)
 - g. Click Import button and navigate to the path where the OMCI_SMS_DEF.mof file is placed
 - h. Select the mof file and click Open



- i. Now we can see the import summary is green without any errors in it. And all the classes are added successfully.
- j. Click on the first radio button (Import both hardware inventory classes and hardware inventory class settings) and Import.

The graphical representation of all the steps is as follows:

Step 1: The following figure displays the hardware inventory setting information in SCCM Console

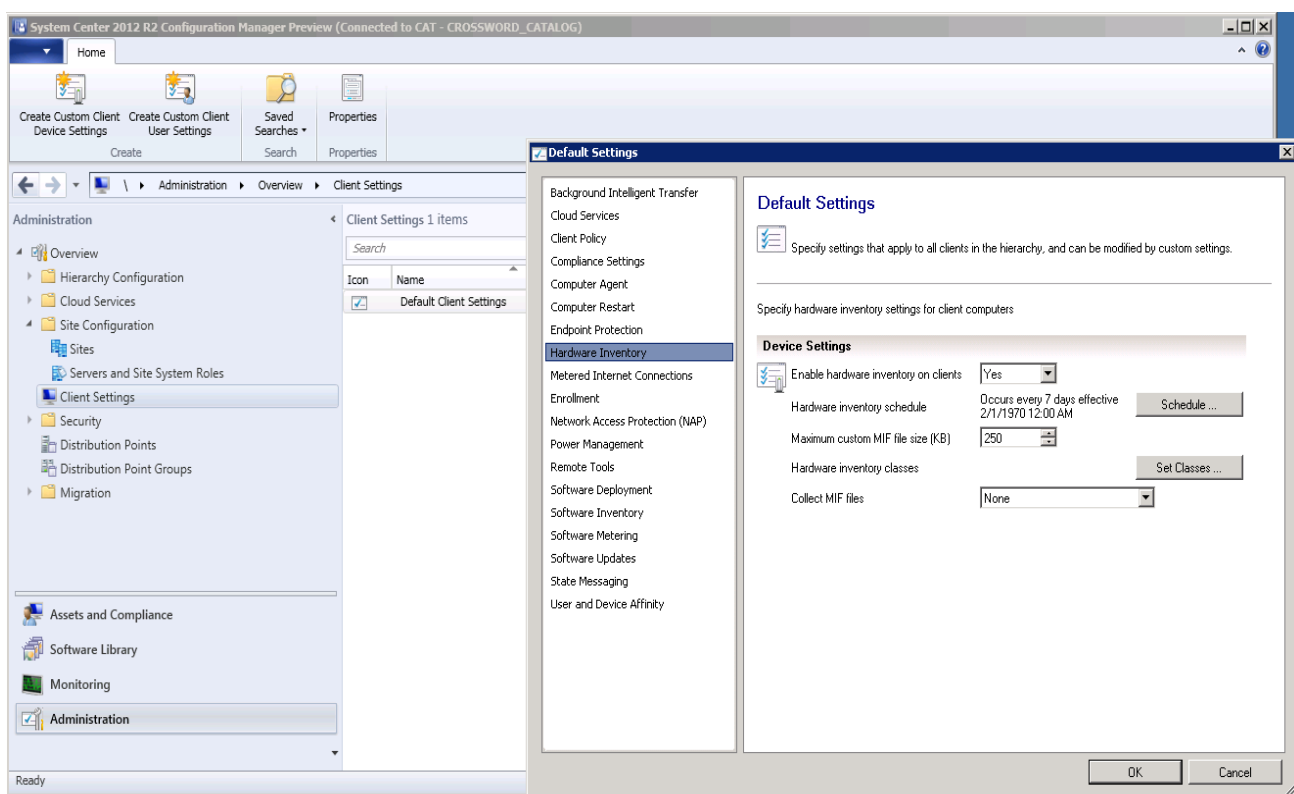


Figure 1 Navigation of the Hardware inventory menu

Step 2: The following figure shows the Hardware inventory classes before importing the DC | M mof file

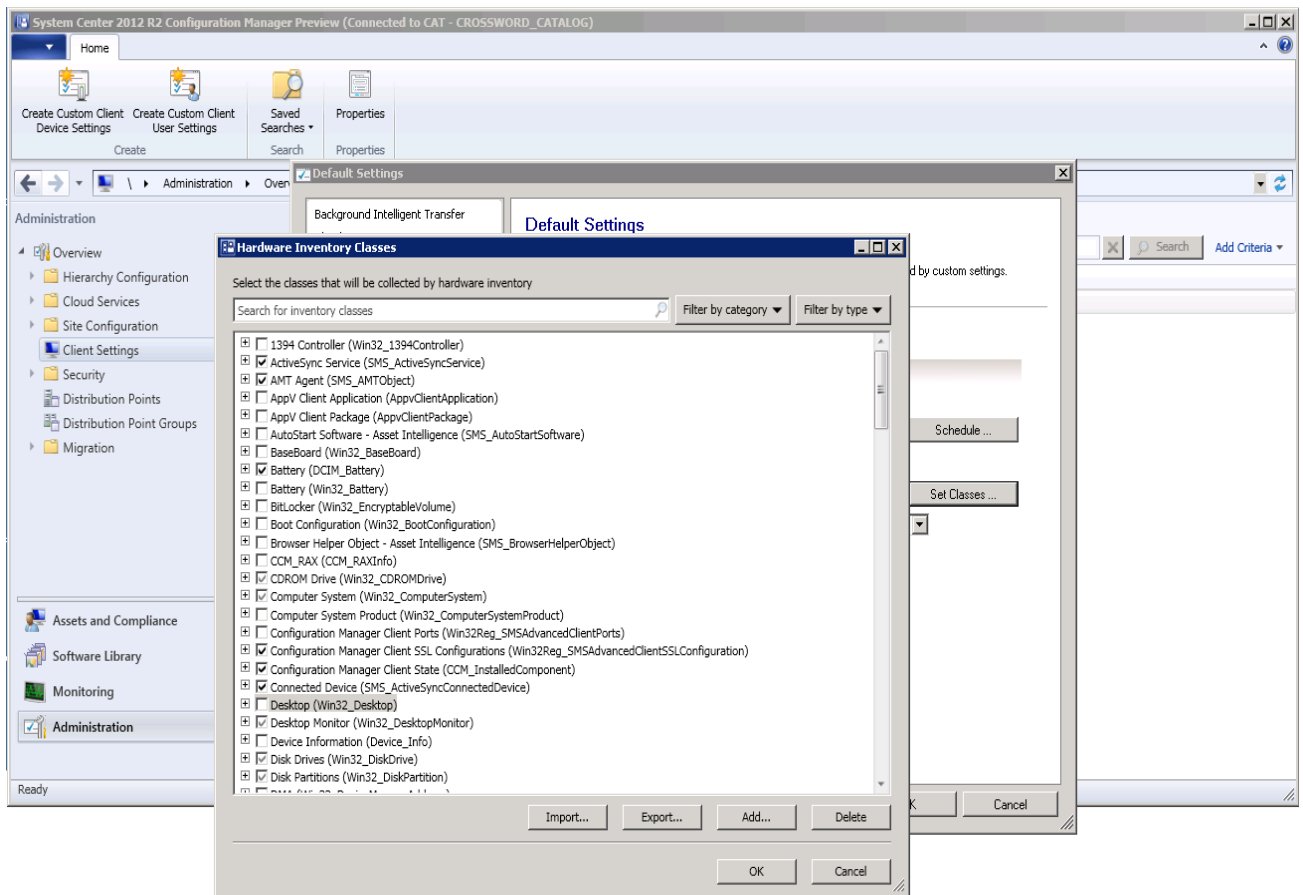


Figure 2 Display of all the classes before importing DCM mof file

Step 3: The following figure displays the importing table for the DC | M mof file.

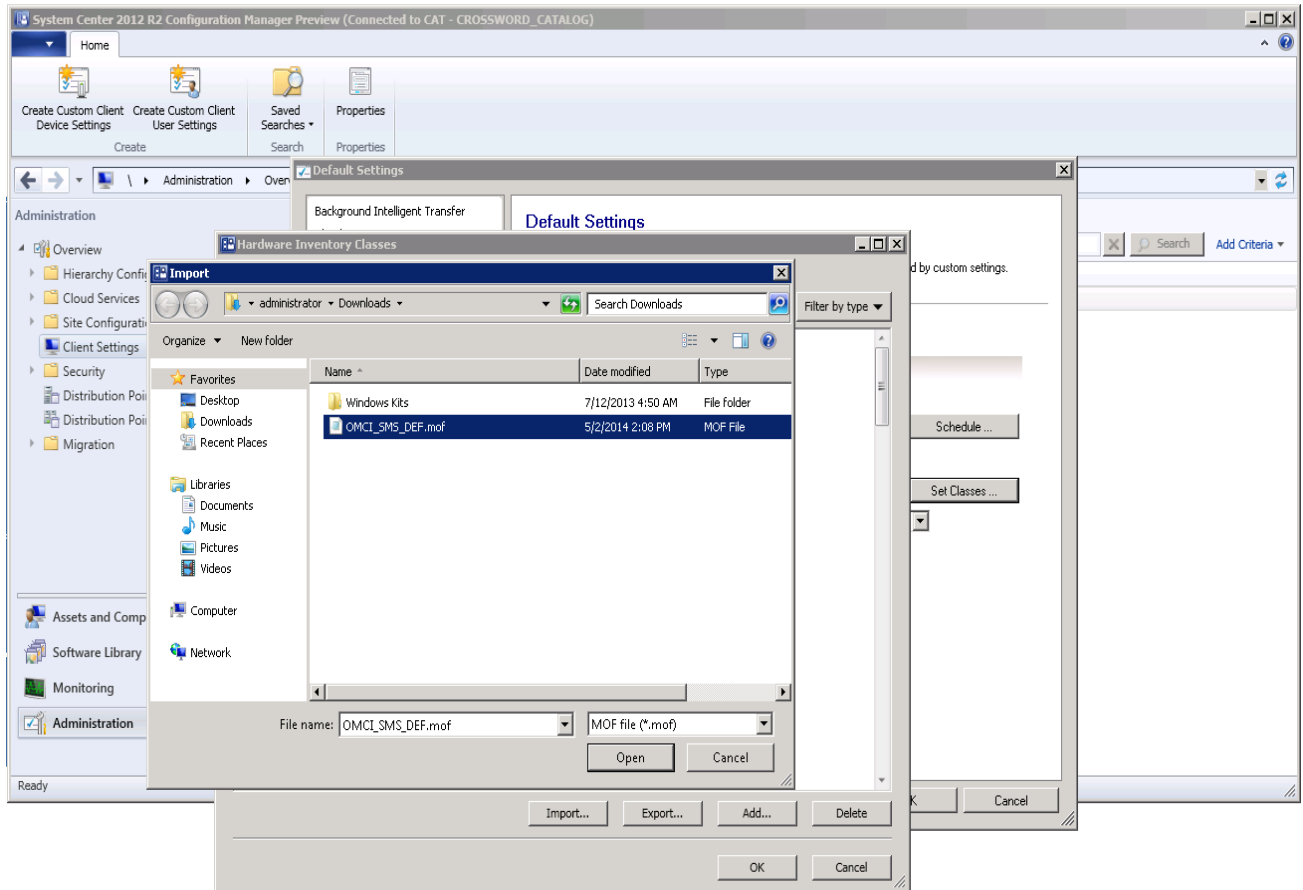


Figure 3 File selection to import the DCM mof file

The screenshot shows the 'Import Summary' dialog box in the System Center 2012 R2 Configuration Manager console. The dialog is titled 'Import Summary' and contains the following text:

Select the inventory items you want to install

- ☒ Import both hardware inventory classes and hardware inventory class settings
- ☐ Import hardware inventory classes only
- ☐ Import hardware inventory class settings only

A green checkmark icon is displayed next to the text: 'The following hardware inventory classes are found in the import file:'

The following classes will be added:

- Battery (DCIM_Battery)
- DCIM Alert Indication Setting Data (DCIM_AlertIndicationSettingData)
- DCIM AMTSettings (DCIM_AMTSettings)
- DCIM ASFSettings (DCIM_ASFSettings)
- DCIM BIOS Enumeration (DCIM_BIOSEnumeration)
- DCIM BIOSElement (DCIM_BIOSElement)
- DCIM BIOSPassword (DCIM_BIOSPassword)
- DCIM BIOSService (DCIM_BIOSService)
- DCIM BIOSString (DCIM_BIOSString)
- DCIM Boot Config Setting (DCIM_BootConfigSetting)
- DCIM Boot Service (DCIM_BootService)
- DCIM Boot Source Setting (DCIM_BootSourceSetting)
- DCIM Button (DCIM_Button)
- DCIM Card (DCIM_Card)
- DCIM Chassis and Docking (DCIM_Chassis)
- DCIM Chip (DCIM_Chip)
- DCIM Computer System (DCIM_ComputerSystem)
- DCIM Concrete Job (DCIM_ConcreteJob)
- DCIM Controller View (DCIM_ControllerView)
- DCIM Desktop Monitor (DCIM_DesktopMonitor)
- DCIM Device Bay (DCIM_DeviceBay)
- DCIM Display Controller (DCIM_DisplayController)
- DCIM Ethernet Port (DCIM_EthernetPort)
- DCIM Fan (DCIM_Fan)

The 'Import' button is highlighted in the dialog box. The background shows the Configuration Manager console with the 'Hardware Inventory Classes' pane selected, displaying a list of classes and their settings.

Step 5: Following the import of the move in ConfigMgr, the display of all the classes in SCCM Console is as follows:

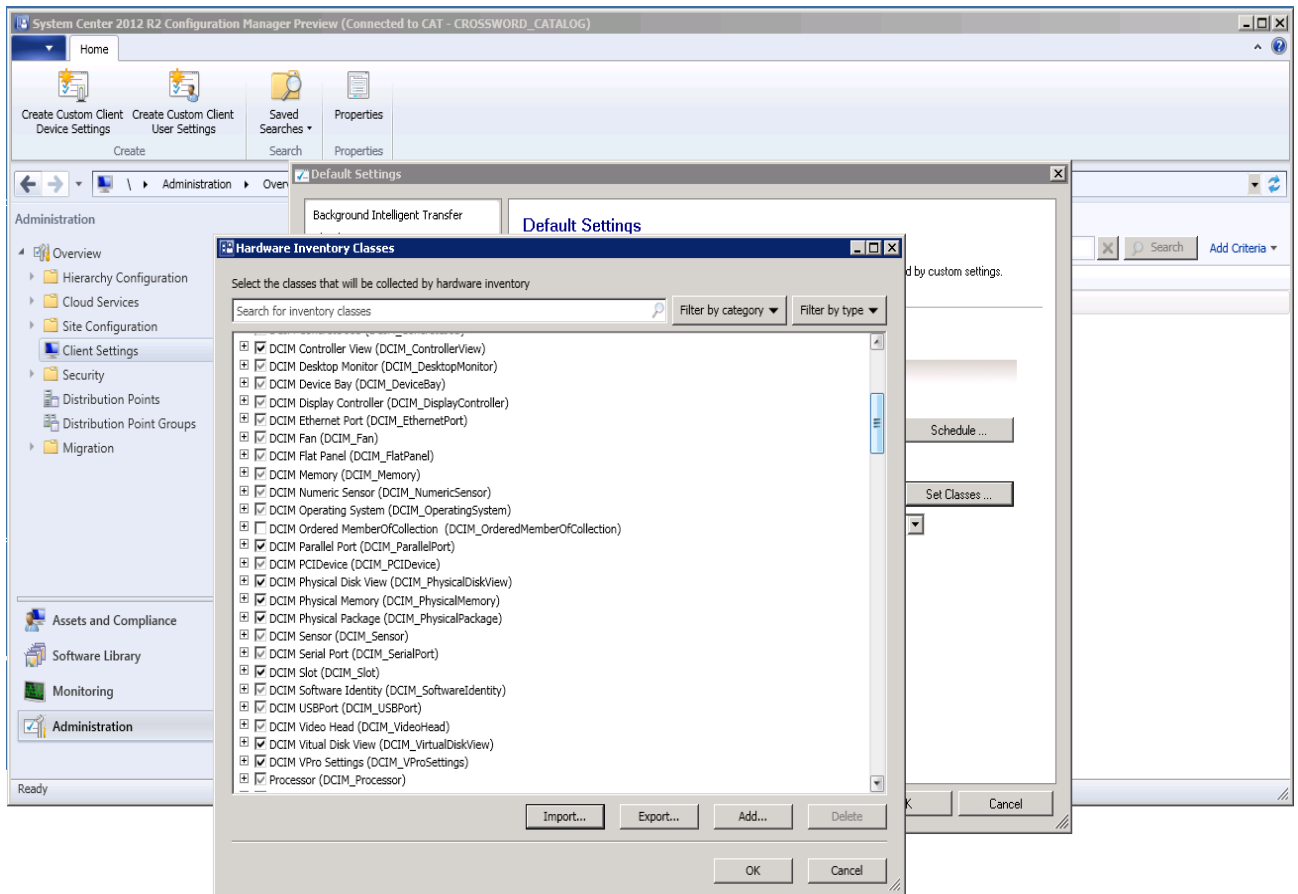


Figure 5 Display of all the classes after importing DCM mof file

2.2.2 Queries Compiled in SCCM

The following are sample reports that may be incorporated into ConfigMgr based on the OMCI SMS MOF in Appendix A. There are two methods that may be used to create new reports within the ConfigMgr console. Reports may be directly created as SQL queries. Alternately, they may be imported using mofcomp against a properly formed MOF file containing the report definition. The following section defines a sample query report both in SQL and MOF formats and explains how each format may be imported.

2.2.2.1 Reported Created as SQL queries

The following is a sample report query for reporting general information on all OMCI deployed commercial systems. The following steps may be used in SCCM 2012 to create a new report from a SQL statement:

The graphical representation of all the steps is as follows:

Step 1: Navigation in SCCM Console to create a SCCM report.

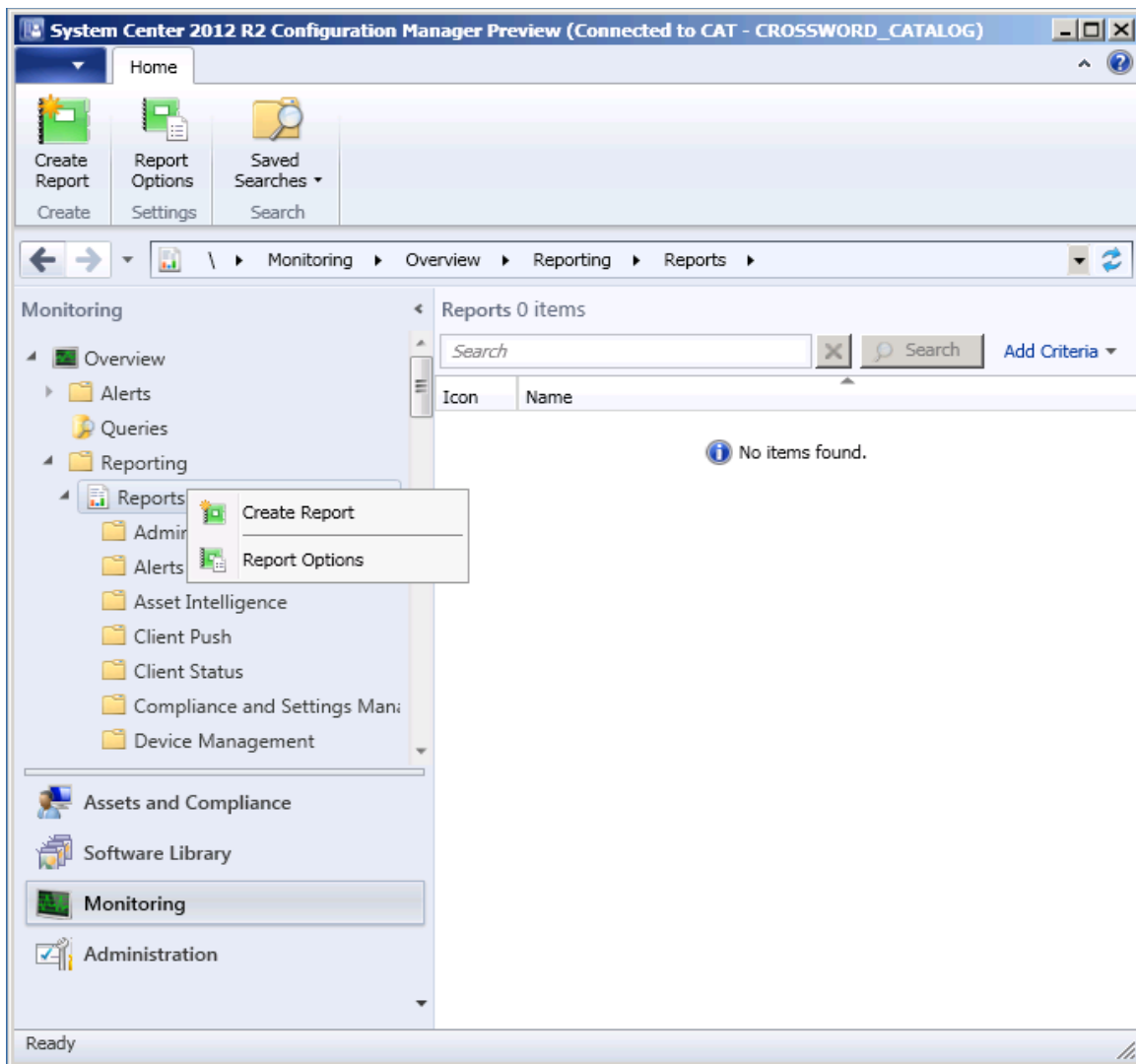


Figure 6 Navigation to create a report for monitoring

Step 2: Update the details in the Report Wizard.

System Center 2012 R2 Configuration Manager Preview (Connected to CAT - CROSSWORD_CATALOG)

Create Report Wizard

Information

Specify the type and location for the report

Select the type of report to be created, then provide a name and description, if needed.

Type

- ☐ Model-based Report
Create a report based on a reporting services model, and select the attributes to be manipulated with Report Builder
- ☒ SQL-based Report
Create a traditional report which is based directly off the database using straight SQL statements and stored procedures

Information

Name:

Description:

Server:

Path:

< Previous Next > Summary Cancel

Figure 7 Input the name and the Path for the SQL based report.

Input the necessary entries here like the name and the path in the wizard tab.

Step 3: Choose the required report data in the SQL Report Builder.

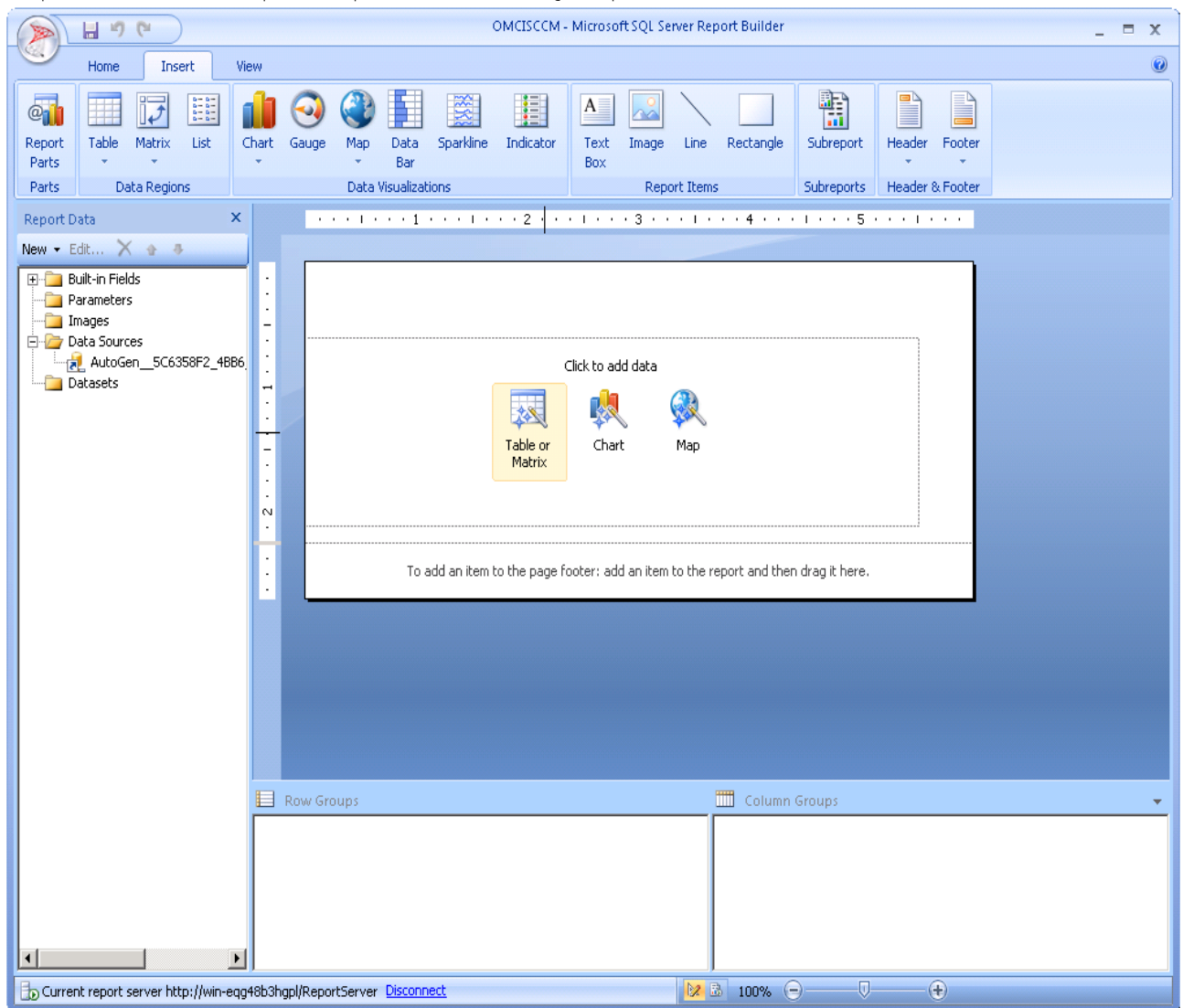


Figure 8 SQL Report Builder display

Step 4: Choose the data set for the report.

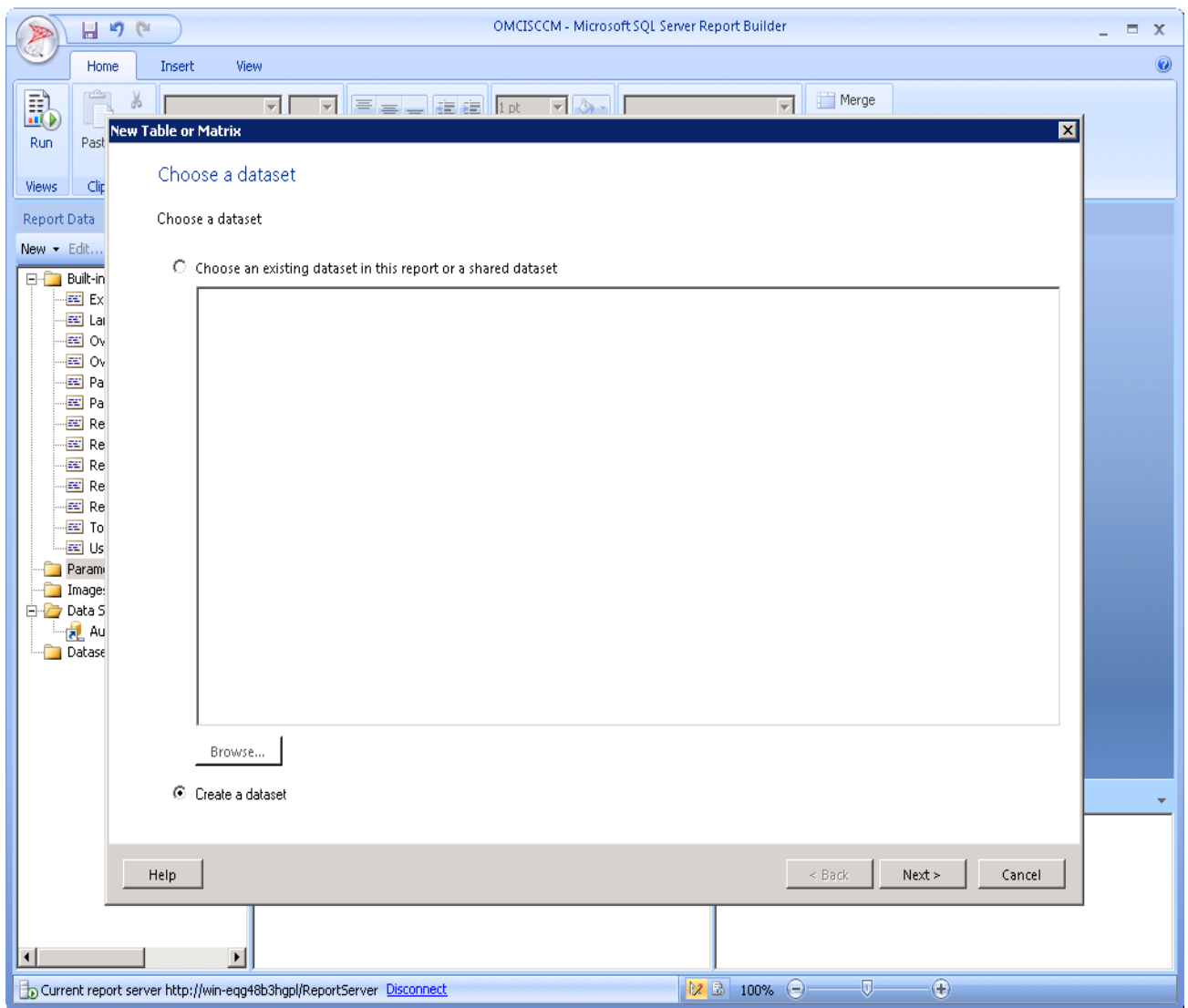


Figure 9 Display of the dataset while creating a new report.

Click Next and it will prompt you to choose the connection to the data source.

Step 5: Input the data source credentials for authentication.

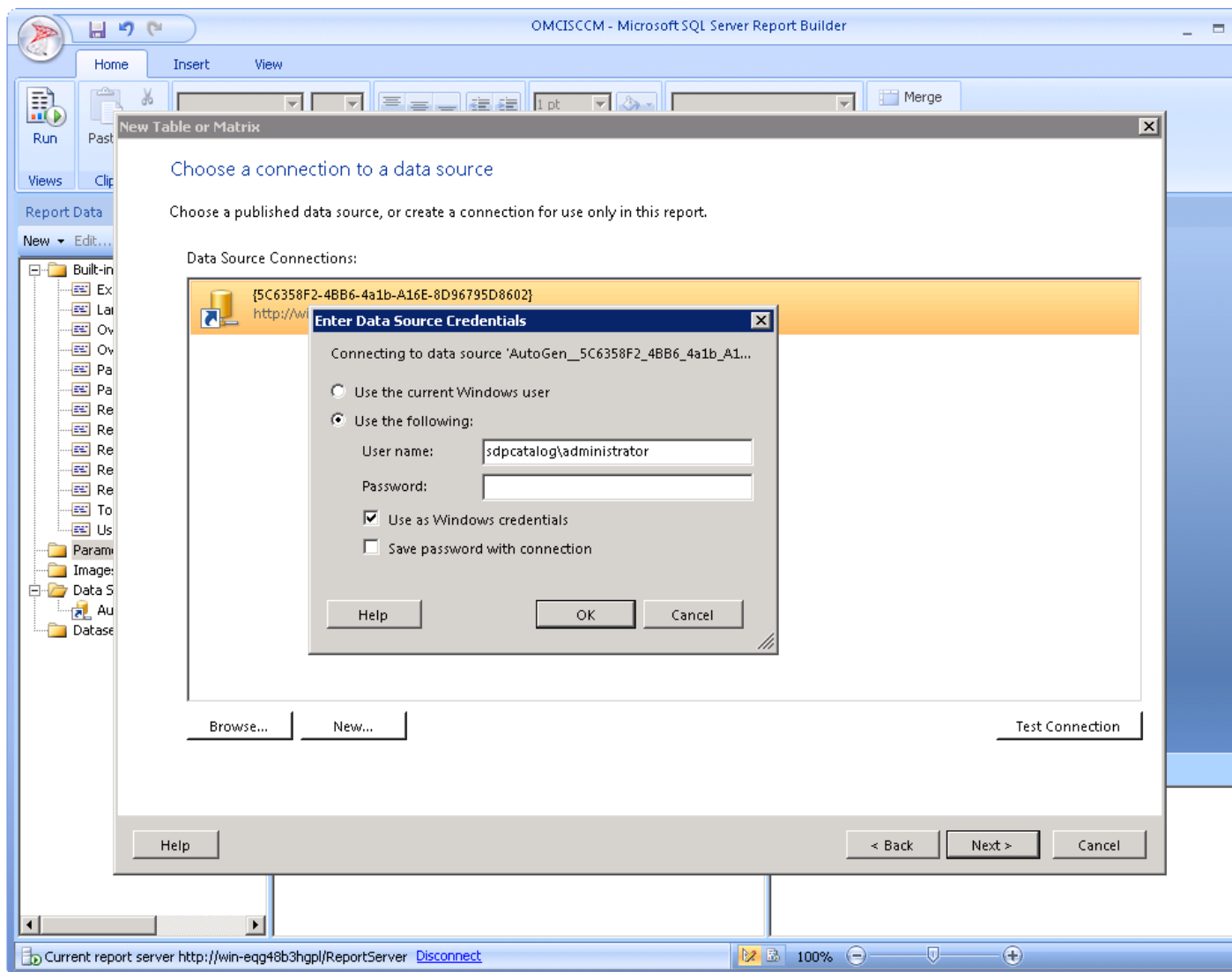


Figure 10 Enter credentials for authentication.

Enter the relevant credentials and click on OK.

Step 6: Update the query to report.

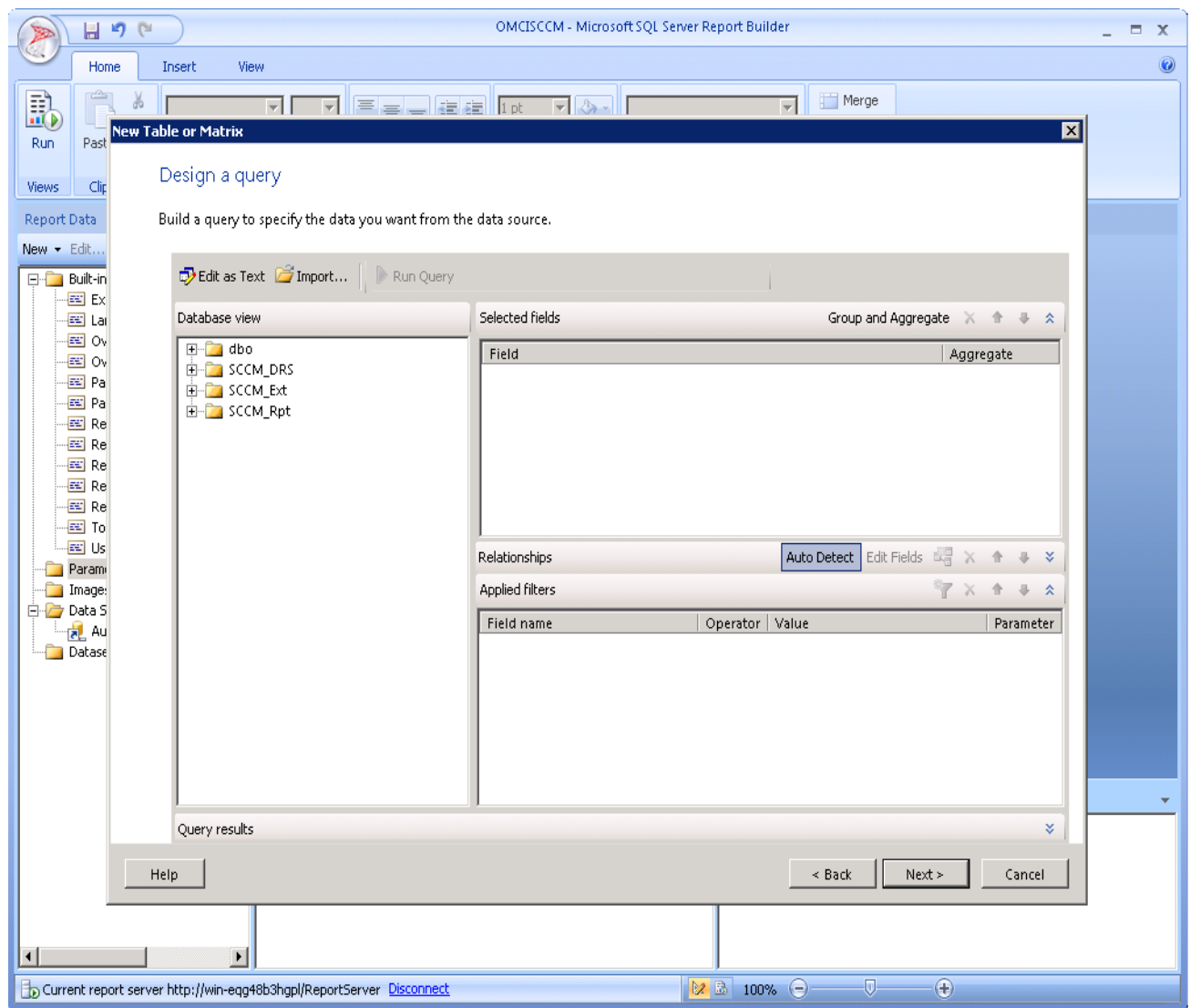


Figure 11 Design Query interface.

Click Edit as Text and paste the following query:

```
-- This sample SQL is provided as an example only, and has not been
-- tested, nor is warranted in any way by Dell; Dell disclaims any
-- liability in connection therewith. Dell provides no technical
-- support with regard to content herein. For more information on
```

-- SQL queries, refer to applicable documentation.

SELECT DSS.ElementName0 as 'Computer Name',

case

when DSS.HealthState0 = 0 then 'Unknown'

when DSS.HealthState0 = 5 then 'OK'

when DSS.HealthState0 = 10 then 'Degraded'

when DSS.HealthState0 = 15 then 'Minor failure'

when DSS.HealthState0 = 20 then 'Major failure'

when DSS.HealthState0 = 25 then 'Critical failure'

when DSS.HealthState0 = 30 then 'Non-recoverable failure'

else CAST(DSS.HealthState0 as varchar)

end as 'Global Health',

DOS.Name0 as 'Operating System', DChas.Model0 as 'Model', DChas.Tag0 as 'Asset Tag',

case

when DChas.ChassisPackageType0 = 1 then 'Unknown'

when DChas.ChassisPackageType0 = 3 then 'Desktop'

when DChas.ChassisPackageType0 = 6 then 'Mini Tower'

when DChas.ChassisPackageType0 = 7 then 'Tower'

when DChas.ChassisPackageType0 = 8 then 'Portable'

when DChas.ChassisPackageType0 = 9 then 'Laptop'

else CAST(DChas.ChassisPackageType0 as varchar)

end as 'System Class',



DBIOS.Version0 as 'BIOS Version'

FROM dbo.v_GS_Dell_ComputerSystem0 as DSS

LEFT JOIN dbo.v_GS_Dell_OperatingSystem0 as DOS on DSS.ResourceID = DOS.ResourceID

LEFT JOIN dbo.v_GS_Dell_Chassis0 as DChas on DSS.ResourceID = DChas.ResourceID

LEFT JOIN dbo.v_GS_Dell_BIOSElement0 as DBIOS on DSS.ResourceID = DBIOS.ResourceID

ORDER BY DSS.ElementName0

Click Next to arrange the fields and choose a layout. After the report created is completed the following display will appear.



Step 7: Arrange the layout.

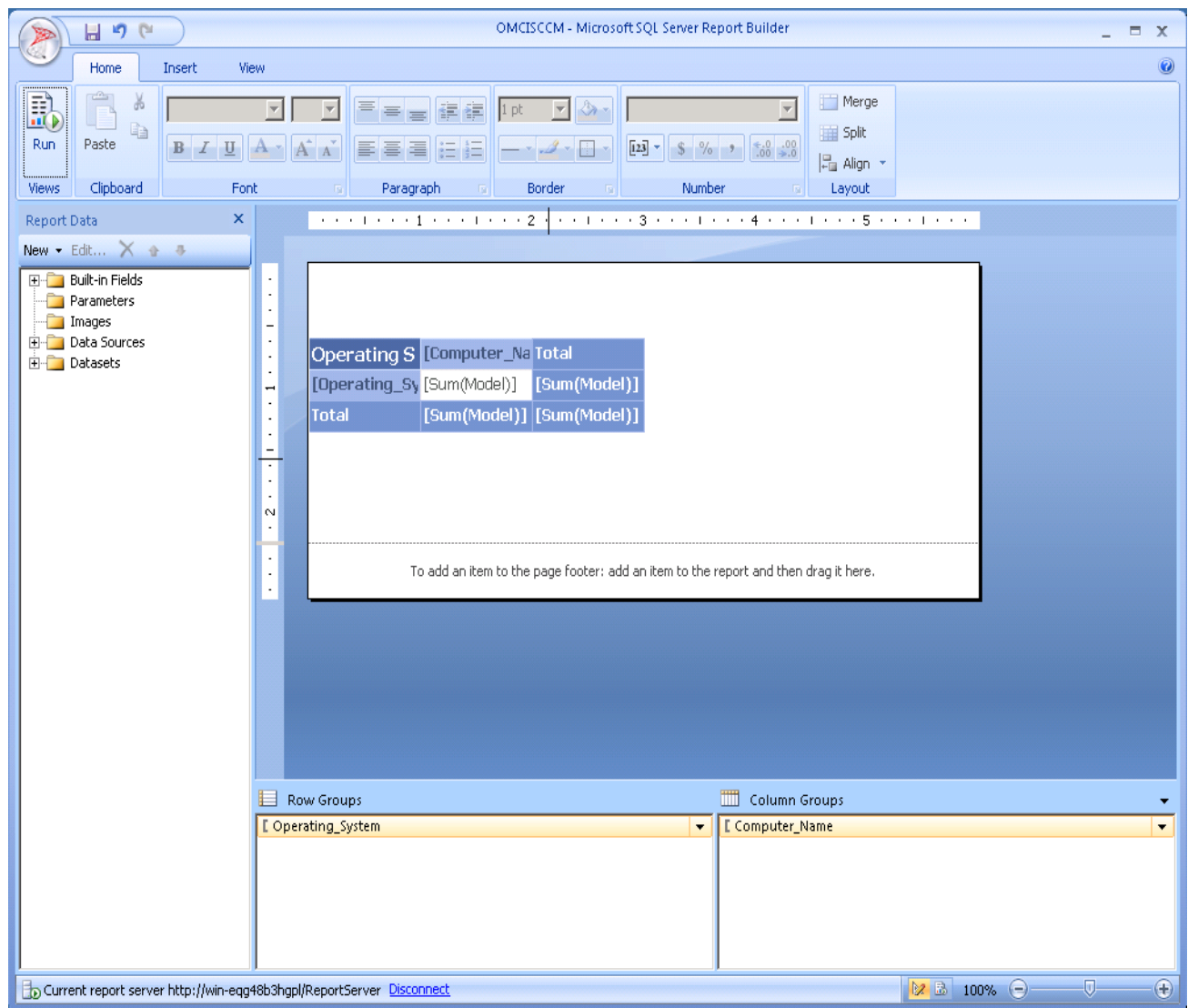


Figure 12 Layout of the submitted query.

Click Run to run the query.

2.2.2.2 Reports created using mofcomp and MOF

The following is a representation of the above report query presented as a MOF that may be (optional) compiled into the ConfigMgr system using mofcomp rather than the SQL query approach. The following steps may be used to import the report using the MOF format:

1. Copy and paste the following text (up to the `“// ***** End *****”` statement) into a local file (e.g. `c:\temp\OMCI_System_Report_Export.MOF`) on the site server system.

2. Then perform a mofcomp of the newly created file. Execute (NOTE: replace "<site_code>" with the actual site code):

```
mofcomp -N:root\SMS\site_<site_code> -instance:createonly  
c:\temp\OMCI_System_Report_Export.MOF
```

```
//-----
```

```
// This sample SQL is provided as an example only, and has not been
```

```
// tested, nor is warranted in any way by Dell; Dell disclaims any
```

```
// liability in connection therewith. Dell provides no technical
```

```
// support with regard to content herein. For more information on
```

```
// SQL queries, refer to applicable documentation.
```

```
//-----
```

```
// ***** Class : SMS_Report *****
```

```
[SecurityVerbs(140551)]
```

```
instance of SMS_Report
```

```
{
```

```
    Category = "Hardware - General";
```

```
    Comment = "List of all Dell OMCI systems";
```

```
    DrillThroughColumns = {};
```

```
    GraphCaption = "";
```

```
    GraphXCol = 1;
```

```
    GraphYCol = 2;
```

```
    MachineDetail = TRUE;
```

```
    MachineSource = FALSE;
```

```
    Name = "All Dell OMCI System List";
```

```
    NumPrompts = 0;
```

```
    RefreshInterval = 0;
```



```

SecurityKey = "";

SQLQuery = "SELECT DSS.ElementName0 as 'Computer Name',

\n

\ncase

\nwhen DSS.HealthState0 = 0 then 'Unknown'

\nwhen DSS.HealthState0 = 5 then 'OK'

\nwhen DSS.HealthState0 = 10 then 'Degraded'

\nwhen DSS.HealthState0 = 15 then 'Minor failure'

\nwhen DSS.HealthState0 = 20 then 'Major failure'

\nwhen DSS.HealthState0 = 25 then 'Critical failure'

\nwhen DSS.HealthState0 = 30 then 'Non-recoverable failure'

\nelse CAST(DSS.HealthState0 as varchar)

\nend as 'Global Health',

\n

\nDOS.Name0 as 'Operating System', DChas.Model0 as 'Model', DChas.Tag0 as 'Asset Tag',

\n

\ncase

\nwhen DChas.ChassisPackageType0 = 1 then 'Unknown'

\nwhen DChas.ChassisPackageType0 = 3 then 'Desktop'

\nwhen DChas.ChassisPackageType0 = 6 then 'Mini Tower'

\nwhen DChas.ChassisPackageType0 = 7 then 'Tower'

\nwhen DChas.ChassisPackageType0 = 8 then 'Portable'

\nwhen DChas.ChassisPackageType0 = 9 then 'Laptop'

\nelse CAST(DChas.ChassisPackageType0 as varchar)

\nend as 'System Class',

\n

```



```
\nDBIOS.Version0 as 'BIOS Version'

\n

\nFROM dbo.v_GS_Dell_ComputerSystem0 as DSS

\nLEFT JOIN dbo.v_GS_Dell_OperatingSystem0 as DOS on DSS.ResourceID = DOS.ResourceID

\nLEFT JOIN dbo.v_GS_Dell_Chassis0 as DChas on DSS.ResourceID = DChas.ResourceID

\nLEFT JOIN dbo.v_GS_Dell_BIOSElement0 as DBIOS on DSS.ResourceID = DBIOS.ResourceID

\nORDER BY DSS.ElementName0";

StatusMessageDetailSource = FALSE;

UnicodeData = FALSE;

XColLabel = "";

YColLabel = "";

};

// ***** End *****
```



A Appendix A

A.1.1 OMCI_SMS_DEF.mof

The OMCI_SMS_DEF.mof file is typically located in the following folder “%ProgramFiles% \Dell\Command Monitor\ssa\omacim”. The contents of the file are as follows:

```
//=====
//
// OMCI_SMS_DEF.mof - Sample MOF that maps SMS inventory set to that
//   provided by the Dell Command | Monitor Provider
//   for Dell Command | Monitor version 9.0.0
//
// ?2008 - 2014 Dell Inc. All rights reserved.
// THIS SOFTWARE IS DISTRIBUTED IN THE HOPE THAT IT WILL BE USEFUL, BUT IS
// PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS, IMPLIED OR OTHERWISE,
// INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTY OF MERCHANTABILITY OR
// FITNESS FOR A PARTICULAR PURPOSE OR ANY WARRANTY REGARDING TITLE OR
// AGAINST INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT,
// INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
// (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTUTE GOODS OR
// SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
// HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
// STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
// IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
// POSSIBILITY OF SUCH DAMAGE.
//
// This sample MOF is provided as an example only. Dell provides no
```



```
// technical support with regard to content herein. For more information

// on MOF files, refer to applicable documentation.

//=====
====

//=====

// Define Dell classes for inventory reporting

//=====

[ SMS_Report(TRUE), SMS_Group_Name("DCIM Alert Indication Setting Data"),
SMS_Class_ID("DELL|DCIM_ALERTINDICATIONSETTINGDATA|1.0"), SMS_Namespace(FALSE),
Namespace("\\\\\\\\\\\\.\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]

class DCIM_AlertIndicationSettingData : SMS_Class_Template
{
[SMS_Report (TRUE)]      string  AlertCategory;
[SMS_Report (TRUE)]      uint16  AlertType;
[SMS_Report (TRUE)]      string  Caption;
[SMS_Report (TRUE)]      uint16  ChangeableType;
[SMS_Report (TRUE)]      string  Description;
[SMS_Report (TRUE)]      string  ElementName;
[SMS_Report (TRUE)]      sint32  EventID;
[SMS_Report (TRUE)]      string  IndicationIdentifier;
[SMS_Report (TRUE), Key] string  InstanceID;
[SMS_Report (TRUE)]      boolean   LocalOnly;
[SMS_Report (TRUE)]      sint32  MaxDisplayNotifications;
[SMS_Report (TRUE)]      sint32  MaxNTEventLogNotifications;

```




```

{
[SMS_Report (TRUE)]      boolean      ASFEnabled;
[SMS_Report (TRUE)]      string   Caption;
[SMS_Report (TRUE)]      uint16   ChangeableType;
[SMS_Report (TRUE)]      string   Description;
[SMS_Report (TRUE)]      string   ElementName;
[SMS_Report (TRUE), Key] string   InstanceID;
[SMS_Report (TRUE)]      uint8    StructureVersion;
};

```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM Battery"), SMS_Class_ID("DELL|DCIM_BATTERY|1.0"),
SMS_Namespace(FALSE), Namespace("\\\\\\\\\\\\\\.\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]

```

```

class DCIM_Battery : SMS_Class_Template

```

```

{
[SMS_Report (TRUE)]      string   Caption;
[SMS_Report (TRUE), Key] string   CreationClassName;
[SMS_Report (TRUE)]      string   Description;
[SMS_Report (TRUE), Key] string   DeviceID;
[SMS_Report (TRUE)]      string   ElementName;
[SMS_Report (TRUE)]      uint16   EnabledDefault;
[SMS_Report (TRUE)]      uint16   EnabledState;
[SMS_Report (TRUE)]      string   ErrorDescription;
[SMS_Report (TRUE)]      uint16   HealthState;
[SMS_Report (TRUE)]      string   IdentifyingDescriptions[];
[SMS_Report (TRUE)]      datetime   InstallDate;
[SMS_Report (TRUE)]      string   Name;

```



[SMS_Report (TRUE)]	uint16	OperationalStatus[];
[SMS_Report (TRUE)]	string	OtherIdentifyingInfo[];
[SMS_Report (TRUE)]	uint16	PrimaryStatus;
[SMS_Report (TRUE)]	uint16	RequestedState;
[SMS_Report (TRUE), Key]	string	SystemCreationClassName;
[SMS_Report (TRUE), Key]	string	SystemName;
[SMS_Report (TRUE)]	uint16	BatteryStatus;
[SMS_Report (TRUE)]	uint16	ChargingStatus;
[SMS_Report (TRUE)]	uint16	Chemistry;
[SMS_Report (TRUE)]	uint32	DesignCapacity;
[SMS_Report (TRUE)]	uint64	DesignVoltage;
[SMS_Report (TRUE)]	uint16	EstimatedChargeRemaining;
[SMS_Report (TRUE)]	uint32	EstimatedRunTime;
[SMS_Report (TRUE)]	uint8	RemainingCapacityMaxError;
[SMS_Report (TRUE)]	string	SmartBatteryVersion;
[SMS_Report (TRUE)]	string	Status;
[SMS_Report (TRUE)]	uint16	TransitioningToState;
		};

```

[SMS_Report (TRUE), Key] string CreationClassName;
[SMS_Report (TRUE), Key] string DeviceID;
[SMS_Report (FALSE)] string ElementName;
[SMS_Report (FALSE)] uint16 EnabledDefault;
[SMS_Report (FALSE)] uint16 EnabledState;
[SMS_Report (FALSE)] uint16 RequestedState;
[SMS_Report (TRUE), Key] string SystemCreationClassName;
[SMS_Report (FALSE), Key] string SystemName;
[SMS_Report (FALSE)] uint16 TransitioningToState;
};

```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM BIOSElement"),
SMS_Class_ID("DELL|DCIM_BIOSELEMENT|1.0"), SMS_Namespace(FALSE),
Namespace("\\\\\\\\\\\\\\.\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]

```

```

class DCIM_BIOSElement : SMS_Class_Template
{
[SMS_Report (FALSE)] string Caption;
[SMS_Report (FALSE)] string Description;
[SMS_Report (FALSE)] string ElementName;
[SMS_Report (FALSE)] uint16 HealthState;
[SMS_Report (FALSE)] datetime InstallDate;
[SMS_Report (TRUE)] string Manufacturer;
[SMS_Report (TRUE), Key] string Name;
[SMS_Report (FALSE)] uint16 OperationalStatus[];
[SMS_Report (FALSE)] uint16 PrimaryStatus;
[SMS_Report (TRUE)] datetime ReleaseDate;
[SMS_Report (FALSE)] string SerialNumber;

```




```

[SMS_Report (FALSE)]      string  ElementName;
[SMS_Report (TRUE), Key]  string  InstanceID;
[SMS_Report (TRUE)]       string  AttributeName;
[SMS_Report (FALSE)]      string  CurrentValue[];
[SMS_Report (TRUE)]       uint32  StringType;
[SMS_Report (TRUE)]       uint64  MaxLength;

```

```
};
```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM Boot Config Setting"),
SMS_Class_ID("DELL|DCIM_BOOTCONFIGSETTING|1.0"), SMS_Namespace(FALSE),
Namespace("\\\\\\\\root\\\\\\dcim\\\\\\sysman") ]

```

```
class DCIM_BootConfigSetting : SMS_Class_Template
```

```
{
```

```
[SMS_Report (TRUE)]      string  ElementName;
```

```
[SMS_Report (TRUE), Key] string  InstanceID;
```

```
};
```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM Boot Service"),
SMS_Class_ID("DELL|DCIM_BOOTSERVICE|1.0"), SMS_Namespace(FALSE),
Namespace("\\\\\\\\root\\\\\\dcim\\\\\\sysman") ]

```

```
class DCIM_BootService : SMS_Class_Template
```

```
{
```

```
[SMS_Report (TRUE), Key] string  CreationClassName;
```

```
[SMS_Report (TRUE), Key] string  Name;
```

```
[SMS_Report (TRUE), Key] string  SystemCreationClassName;
```

```
[SMS_Report (TRUE), Key] string  SystemName;
```



```
};
```

```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Boot Source Setting"),  
SMS_Class_ID("DELL|DCIM_BOOTSOURCESETTING|1.0"), SMS_Namespace(FALSE),  
Namespace("\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\root\\\\\\\\\\\\dcim\\\\\\\\\\\\sysman") ]
```

```
class DCIM_BootSourceSetting : SMS_Class_Template
```

```
{
```

```
[SMS_Report (TRUE)]      string  BIOSBootString;  
[SMS_Report (TRUE)]      string  Caption;  
[SMS_Report (TRUE)]      uint16  ChangeableType;  
[SMS_Report (TRUE)]      string  Description;  
[SMS_Report (TRUE)]      string  ElementName;  
[SMS_Report (TRUE)]      uint16  FailThroughSupported;  
[SMS_Report (TRUE), Key]  string  InstanceID;  
[SMS_Report (TRUE)]      string  StructuredBootString;
```

```
};
```

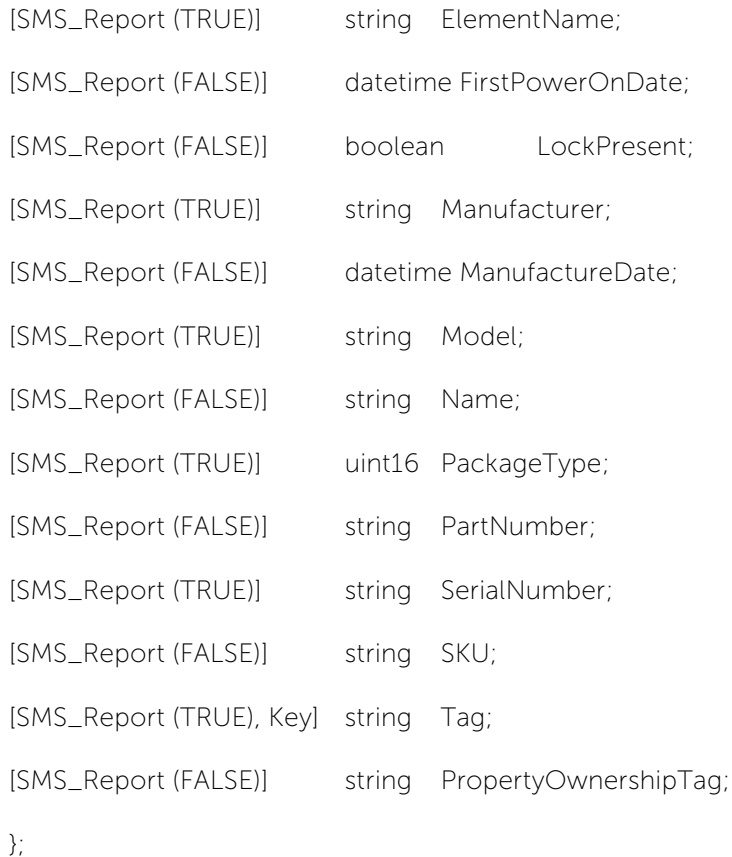
```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Card"), SMS_Class_ID("DELL|DCIM_CARD|1.0"),  
SMS_Namespace(FALSE), Namespace("\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\root\\\\\\\\\\\\dcim\\\\\\\\\\\\sysman") ]
```

```
class DCIM_Card : SMS_Class_Template
```

```
{
```

```
[SMS_Report (TRUE)]      boolean      CanBeFRUed;  
[SMS_Report (TRUE)]      string  Caption;  
[SMS_Report (TRUE), Key]  string  CreationClassName;  
[SMS_Report (TRUE)]      string  Description;  
[SMS_Report (TRUE)]      string  ElementName;  
[SMS_Report (TRUE)]      uint16  HealthState;
```





```

[SMS_Report (TRUE)]      uint16  IRQ;
[SMS_Report (TRUE)]      datetime      ManufactureDate;
[SMS_Report (TRUE)]      string  Manufacturer;
[SMS_Report (TRUE)]      string  Model;
[SMS_Report (TRUE)]      string  Name;
[SMS_Report (TRUE)]      uint16  OperationalStatus[];
[SMS_Report (TRUE)]      string  OtherIdentifyingInfo;
[SMS_Report (TRUE)]      string  PartNumber;
[SMS_Report (TRUE)]      uint16  PrimaryStatus;
[SMS_Report (TRUE)]      string  SerialNumber;
[SMS_Report (TRUE)]      string  SKU;
[SMS_Report (TRUE), Key] string  Tag;
[SMS_Report (TRUE)]      string  Version;
};

```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM Concrete Job"),
SMS_Class_ID("DELL|DCIM_CONCRETEJOB|1.0"), SMS_Namespace(FALSE),
Namespace("\\\\\\\\\\\\\\.\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]

```

```

class DCIM_ConcreteJob : SMS_Class_Template
{
[SMS_Report (FALSE)]      string  Caption;
[SMS_Report (FALSE)]      boolean      DeleteOnCompletion;
[SMS_Report (FALSE)]      string  Description;
[SMS_Report (FALSE)]      string  ElementName;
[SMS_Report (TRUE)]       uint16  ErrorCode;
[SMS_Report (TRUE)]       string  ErrorDescription;
[SMS_Report (FALSE)]      uint16  HealthState;

```






```

[SMS_Report (TRUE)]      string    EDIDVersionNumber;
[SMS_Report (TRUE)]      boolean      InputAnalog;
[SMS_Report (TRUE)]      boolean      InputDigital;
[SMS_Report (TRUE)]      boolean      InputDisplayPort;
[SMS_Report (TRUE)]      boolean      InputDVI;
[SMS_Report (TRUE)]      uint16    InputSource;
[SMS_Report (TRUE)]      string    ManufactureDate;
[SMS_Report (TRUE)]      uint32    MaxBrightness;
[SMS_Report (TRUE)]      uint32    MaxContrast;
[SMS_Report (TRUE)]      uint64    MaxQuiesceTime;
[SMS_Report (TRUE)]      string    MCCSVersionNumber;
[SMS_Report (TRUE)]      uint16    PhysicalSizeH;
[SMS_Report (TRUE)]      uint16    PhysicalSizeV;
[SMS_Report (TRUE)]      uint16    PrimaryStatus;
[SMS_Report (TRUE)]      string    ProductCode;
[SMS_Report (TRUE)]      string    SerialNumber;
[SMS_Report (TRUE)]      boolean      SeparateSyncHVSuported;
[SMS_Report (TRUE)]      boolean      sRGBStandardDefaultColorSpace;
[SMS_Report (TRUE)]      uint16    StandbyModeSupported;
[SMS_Report (TRUE)]      uint16    SuspendModeSupported;
[SMS_Report (TRUE), Key] string    SystemCreationClassName;
[SMS_Report (TRUE), Key] string    SystemName;
[SMS_Report (TRUE)]      uint16    VeryLowPowerSupported;
};

```



```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Device Bay"),  
SMS_Class_ID("DELL|DCIM_DEVICEBAY|1.0"), SMS_Namespace(FALSE),  
Namespace("\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\root\\\\\\\\\\\\\\\\dcim\\\\\\\\\\\\\\\\sysman") ]
```

```
class DCIM_DeviceBay : SMS_Class_Template
```

 $\{$

```
[SMS_Report (FALSE)]    string    Caption;
```

```
[SMS_Report(TRUE), Key]    string    CreationClassName;
```

```
[SMS_Report (FALSE)]      string    Description;
```

uint16 DeviceBayType;

```
[SMS_Report (TRUE)]      string    DeviceBayLocation;
```

```
[SMS_Report (TRUE)]      string    DeviceCurrentlyAttached;
```

```
[SMS_Report (TRUE), Key] string DeviceID;
```

```
[SMS_Report (TRUE)]      string    DeviceSupported;
```

```
[SMS_Report (TRUE)]      string      ElementName;
```

uint16 EnabledDefault;

```
[SMS_Report (TRUE)]      uint16  EnabledState;
```

```
[SMS_Report (FALSE)]      string  ErrorDescription;
```

uint16 HealthState;

```
[SMS_Report (TRUE)]      string    IdentifyingDescriptions[];
```

[SMS_Report (FALSE)] datetime InstallDate;

```
[SMS_Report (TRUE)]      uint16  LogicalModuleType;
```

uint16 ModuleNumber;

```
[SMS_Report (FALSE)]    string    Name;
```

uint16 OperationalStatus[];

```
[SMS_Report (TRUE)]      string  OtherIdentifyingInfo[];
```

uint16 PrimaryStatus;

uint16 RequestedState;




```
[SMS_Report (TRUE), Key] string SystemName;

[SMS_Report (FALSE)] uint16 TOEEnabledState;

[SMS_Report (FALSE)] uint16 TransitioningToState;

};
```

```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Fan"), SMS_Class_ID("DELL|DCIM_FAN|1.0"),
SMS_Namespace(FALSE), Namespace("\\\\\\\\\\\\\\.\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]
```

```
class DCIM_Fan : SMS_Class_Template
```

```
{

[SMS_Report (TRUE)] boolean ActiveCooling;

[SMS_Report (FALSE)] string Caption;

[SMS_Report (TRUE), Key] string CreationClassName;

[SMS_Report (FALSE)] string Description;

[SMS_Report (TRUE), Key] string DeviceID;

[SMS_Report (TRUE)] string ElementName;

[SMS_Report (TRUE)] uint16 EnabledDefault;

[SMS_Report (TRUE)] uint16 EnabledState;

[SMS_Report (TRUE)] string ErrorDescription;

[SMS_Report (FALSE)] boolean ErrorCleared;

[SMS_Report (TRUE)] uint16 HealthState;

[SMS_Report (FALSE)] string IdentifyingDescriptions[];

[SMS_Report (FALSE)] datetime InstallDate;

[SMS_Report (TRUE)] string Name;

[SMS_Report (TRUE)] uint16 OperationalStatus[];

[SMS_Report (TRUE)] uint16 PrimaryStatus;

[SMS_Report (TRUE)] uint16 RequestedState;
```



```
[SMS_Report (TRUE), Key] string SystemCreationClassName;
[SMS_Report (TRUE), Key] string SystemName;
[SMS_Report (TRUE)] boolean VariableSpeed;
[SMS_Report (TRUE)] uint16 TransitioningToState;
};
```

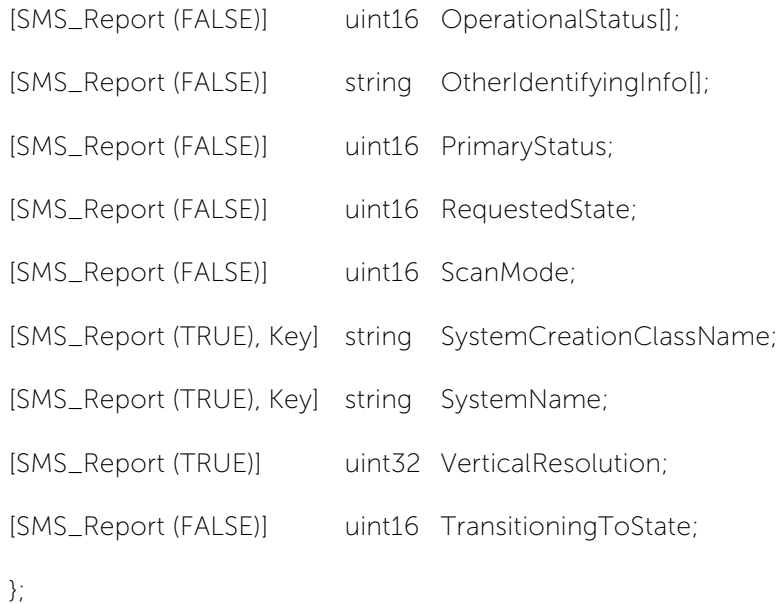
```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Flat Panel"), SMS_Class_ID("DELL|DCIM_FLATPANEL|1.0"),
SMS_Namespace(FALSE), Namespace("\\\\\\\\\\\\\\.\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]
```

```
class DCIM_FlatPanel : SMS_Class_Template
```

```
{
```

```
[SMS_Report (TRUE)] string Brightness;
[SMS_Report (FALSE)] string Caption;
[SMS_Report (TRUE), Key] string CreationClassName;
[SMS_Report (FALSE)] string Description;
[SMS_Report (TRUE), Key] string DeviceID;
[SMS_Report (TRUE)] uint16 DisplayType;
[SMS_Report (TRUE)] string ElementName;
[SMS_Report (TRUE)] uint16 EnabledDefault;
[SMS_Report (TRUE)] uint16 EnabledState;
[SMS_Report (FALSE)] string ErrorDescription;
[SMS_Report (FALSE)] uint16 HealthState;
[SMS_Report (TRUE)] uint32 HorizontalResolution;
[SMS_Report (FALSE)] uint64 MaxQuiesceTime;
[SMS_Report (FALSE)] string IdentifyingDescriptions[];
[SMS_Report (FALSE)] datetime InstallDate;
[SMS_Report (FALSE)] string Name;
```

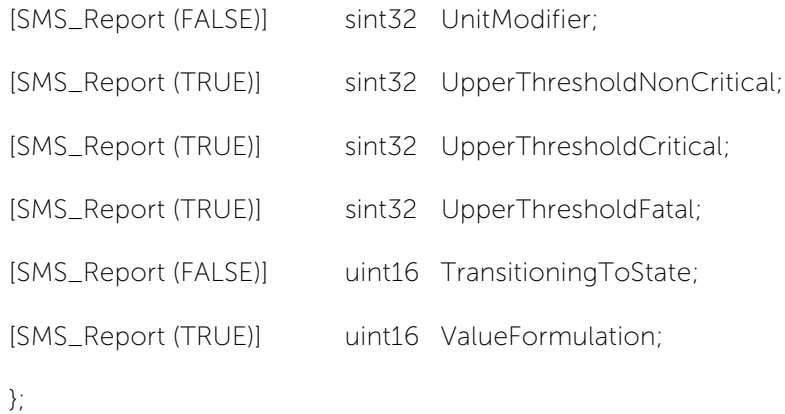




[SMS_Report (FALSE)]	uint16	ErrorInfo;
[SMS_Report (TRUE)]	string	ErrorMethodology;
[SMS_Report (TRUE)]	uint16	FailOverState;
[SMS_Report (TRUE)]	uint16	HealthState;
[SMS_Report (FALSE)]	string	IdentifyingDescriptions[];
[SMS_Report (TRUE)]	boolean	Primordial;
[SMS_Report (FALSE)]	datetime	InstallDate;
[SMS_Report (TRUE)]	uint64	NumberOfBlocks;
[SMS_Report (TRUE)]	uint16	OperationalStatus[];
[SMS_Report (TRUE)]	string	OtherIdentifyingInfo[];
[SMS_Report (TRUE)]	uint16	PrimaryStatus;
[SMS_Report (TRUE)]	string	Purpose;
[SMS_Report (TRUE)]	uint16	RedundancyConfiguration;
[SMS_Report (TRUE)]	uint16	RequestedState;
[SMS_Report (TRUE), Key]	string	SystemCreationClassName;
[SMS_Report (TRUE), Key]	string	SystemName;
[SMS_Report (TRUE)]	boolean	volatile;
[SMS_Report (TRUE)]	uint16	TransitioningToState;
		};

[SMS_Report (TRUE), Key]	string	CreationClassName;
[SMS_Report (FALSE)]	sint32	CurrentReading;
[SMS_Report (TRUE)]	string	CurrentState;
[SMS_Report (FALSE)]	string	Description;
[SMS_Report (TRUE), Key]	string	DeviceID;
[SMS_Report (TRUE)]	string	ElementName;
[SMS_Report (TRUE)]	uint16	EnabledDefault;
[SMS_Report (TRUE)]	uint16	EnabledState;
[SMS_Report (FALSE)]	string	ErrorDescription;
[SMS_Report (TRUE)]	uint16	HealthState;
[SMS_Report (FALSE)]	string	IdentifyingDescriptions[];
[SMS_Report (FALSE)]	datetime	InstallDate;
[SMS_Report (TRUE)]	sint32	LowerThresholdNonCritical;
[SMS_Report (TRUE)]	sint32	LowerThresholdFatal;
[SMS_Report (TRUE)]	sint32	LowerThresholdCritical;
[SMS_Report (FALSE)]	string	Name;
[SMS_Report (TRUE)]	uint16	OperationalStatus[];
[SMS_Report (FALSE)]	string	OtherIdentifyingInfo[];
[SMS_Report (FALSE)]	uint64	PollingInterval;
[SMS_Report (TRUE)]	string	PossibleStates[];
[SMS_Report (FALSE)]	uint16	PrimaryStatus;
[SMS_Report (TRUE)]	uint16	RateUnits;
[SMS_Report (FALSE)]	uint16	RequestedState;
[SMS_Report (TRUE)]	uint16	SensorType;
[SMS_Report (TRUE), Key]	string	SystemCreationClassName;
[SMS_Report (TRUE), Key]	string	SystemName;








```

class DCIM_PhysicalPackage : SMS_Class_Template
{
[SMS_Report (TRUE)]      boolean      CanBeFRUed;
[SMS_Report (TRUE)]      string  Caption;
[SMS_Report (TRUE), Key] string  CreationClassName;
[SMS_Report (TRUE)]      string  Description;
[SMS_Report (TRUE)]      string  ElementName;
[SMS_Report (TRUE)]      uint16   HealthState;
[SMS_Report (TRUE)]      datetime   InstallDate;
[SMS_Report (TRUE)]      datetime   ManufactureDate;
[SMS_Report (TRUE)]      string  Manufacturer;
[SMS_Report (TRUE)]      string  Model;
[SMS_Report (TRUE)]      string  Name;
[SMS_Report (TRUE)]      uint16   OperationalStatus[];
[SMS_Report (TRUE)]      string  OtherIdentifyingInfo;
[SMS_Report (TRUE)]      string  OtherPackageType;
[SMS_Report (TRUE)]      uint16   PackageType;
[SMS_Report (TRUE)]      string  PartNumber;
[SMS_Report (TRUE)]      uint16   PrimaryStatus;
[SMS_Report (TRUE)]      string  SerialNumber;
[SMS_Report (TRUE)]      string  SKU;
[SMS_Report (TRUE), Key] string  Tag;
[SMS_Report (TRUE)]      string  VendorCompatibilityStrings[];
[SMS_Report (TRUE)]      string  Version;
};

```



```
[ SMS_Report(TRUE), SMS_Group_Name("DCIM Sensor"), SMS_Class_ID("DELL|DCIM_SENSOR|1.0"),  
SMS_Namespace(FALSE), Namespace("\\\\\\\\\\\\\\.\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]
```

```
class DCIM_Sensor : SMS_Class_Template
```

 $\{$

```
[SMS_Report (FALSE)]    string    Caption;
```

```
[SMS_Report(TRUE), Key]    string    CreationClassName;
```

```
[SMS_Report (TRUE)]      string    CurrentState;
```

```
[SMS_Report (FALSE)]      string    Description;
```

```
[SMS_Report (TRUE), Key] string DeviceID;
```

```
[SMS_Report (TRUE)]      string  ElementName;
```

uint16 EnabledDefault;

```
[SMS_Report (FALSE)]      uint16  EnabledState;
```

```
[SMS_Report (FALSE)]    string    ErrorDescription;
```

uint16 HealthState;

```
[SMS_Report (FALSE)]      string    IdentifyingDescriptions[];
```

[SMS_Report (FALSE)] datetime InstallDate;

```
[SMS_Report (FALSE)]    string    Name;
```

uint16 OperationalStatus[];

```
[SMS_Report (FALSE)]      string    OtherIdentifyingInfo[];
```

uint64 PollingInterval;

```
[SMS_Report (TRUE)]      string    PossibleStates[];
```

```
[SMS_Report (FALSE)]      uint16  PrimaryStatus;
```

uint16 RequestedState;

uint16 SensorType;

[SMS_Report (TRUE), Key] string SystemCreationClassName;

```
[SMS_Report (TRUE), Key] string SystemName;
```







```

[SMS_Report (TRUE)]      string  ElementName;
[SMS_Report (FALSE)]     uint16  HealthState;
[SMS_Report (TRUE)]      string  IdentityInfoType[];
[SMS_Report (TRUE)]      string  IdentityInfoValue[];
[SMS_Report (FALSE)]     datetime  InstallDate;
[SMS_Report (TRUE), Key] string  InstanceID;
[SMS_Report (TRUE)]      boolean   IsEntity;
[SMS_Report (TRUE)]      string  Manufacturer;
[SMS_Report (TRUE)]      string  Name;
[SMS_Report (FALSE)]     uint16  OperationalStatus[];
[SMS_Report (FALSE)]     uint16  PrimaryStatus;
[SMS_Report (TRUE)]      datetime  ReleaseDate;
[SMS_Report (FALSE)]     string  SerialNumber;
[SMS_Report (TRUE)]      string  TargetTypes[];
[SMS_Report (TRUE)]      string  VersionString;
[SMS_Report (FALSE)]     uint16  TransitioningToState;
[SMS_Report (FALSE)]     uint16  RevisionNumber;
};

```

```

[ SMS_Report(TRUE), SMS_Group_Name("DCIM USBPort"), SMS_Class_ID("DELL|DCIM_USBPORT|1.0"),
SMS_Namespace(FALSE), Namespace("\\\\\\\\root\\\\\\\\dcim\\\\\\\\sysman") ]

```

```

class DCIM_USBPort : SMS_Class_Template
{
[SMS_Report (FALSE)]      string  Caption;
[SMS_Report (TRUE), Key] string  CreationClassName;
[SMS_Report (FALSE)]      string  Description;

```







B Additional resources

Support.dell.com is focused on meeting your needs with proven services and support.

DellTechCenter.com is an IT Community where you can connect with Dell customers and Dell employees for the purpose of sharing knowledge, best practices, and information about Dell products and installations.

Referenced or recommended Dell publications:

- Install"Guide":
- User"Guide":
- Reference"Guide
- SNMP"Reference"Guide":
http://downloads.dell.com/published/Pages/index.html#esuprt_client_sys_mgmt_opnmang_clnt_instr
- Extending"Client"Instrumentation"using"OMCI
http://en.community.dell.com/techcenter/extras/m/white_papers/20410895.aspx
- Prompting"collection"to"report"in"SCCM"2012
*****<http://asithadesilva.wordpress.com/2013/04/01/how-to-prompt-collection-to-a-report-in-sccm-2012/>

