

Operations Manager 2007 Report Authoring Guide

Microsoft Corporation

Published: September 2008

Authors

Daniel Savage, Liza Poggemeyer

Feedback

Send suggestions and comments about this document to [momdocs@microsoft.com](mailto:momdocs@microsoft.com?subject=Report%20%20Authoring%20Guide%20published%20Feb%202008). Please include the Report Authoring Guide name and publish date with your feedback.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Server, Windows Vista, and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Revision History

|  |  |
| --- | --- |
| Release Date | Changes |
| February 26, 2008 | Original release of this guide |
| February 29, 2008 | Corrected the syntax of the query to create a custom report |
| April 11, 2008 | Clarified the procedures available in the scenario to create and publish a report. |
| September 30, 2008 | Added information about publishing reports to a SharePoint site, as well as improved the information about scheduling reports. |

Contents

[Operations Manager 2007 Report Authoring Guide 5](#_Toc230671030)

[Overview of the Reporting Feature 5](#_Toc230671031)

[Working with Reports 9](#_Toc230671032)

[Publishing a Generic Report 10](#_Toc230671033)

[Publishing the Availabiity Report 10](#_Toc230671034)

[Sharing the Availability Report 11](#_Toc230671035)

[Publishing the Availability Report to Sharepoint 12](#_Toc230671036)

[Using Linked Reports 13](#_Toc230671037)

[Types of Linked Reports 13](#_Toc230671038)

[Creating a Linked Report 14](#_Toc230671039)

[Authoring Custom Reports 19](#_Toc230671040)

[Custom Reports - Sample Scenario for a Simple Report 1 20](#_Toc230671041)

[Custom Reports - Sample Scenario for a Simple Report 2 23](#_Toc230671042)

[Custom Reports - Sample Scenario for an Advanced Report 25](#_Toc230671043)

[Scheduling and Sending Reports 38](#_Toc230671044)

[Creating a Report Schedule 38](#_Toc230671045)

[Sending a Report Through E-mail 39](#_Toc230671046)

[Delivering a report to the SQL Server Report Server Cache 40](#_Toc230671047)

[Appendix A - Reporting Management Pack Schema 41](#_Toc230671048)

[Appendix B - Parameter Values and Parameter Controls 44](#_Toc230671049)

[Appendix C - Data Types and Sample Queries 64](#_Toc230671050)

Operations Manager 2007 Report Authoring Guide

Operations Manager 2007 collects large amounts of data from your environment. By using the Reporting feature, you can create reports based on this data that provide additional information about the health of your environment. For example, you can create a report that shows the amount of time it takes between an alert being raised to its being written in the Operations Manager database. This report can help you identify any network delays and isolate trouble spots. Based on the data in this report, you can then take corrective action.

Before you can use the Reporting feature, you need to install the required reporting components, as described in the Operations Manager Deployment Guide. You can also find additional information about using the Reporting interface in the Operations Manager Help.

The following sections provide an introduction to the Reporting feature, including a description of changes from the previous release of Operations Manager.

Overview of the Reporting Feature

Operations Manager Reporting is based on Microsoft SQL Server 2005 and on SQL Reporting Services. The feature includes the following components:

 Databases—OperationsManager and OperationsManagerDW

 Data Warehouse Server

 Management server

 Operations Manager Reporting Server

 Reports

 Management packs

Databases

Reporting uses the following databases:

 OperationsManager

This database contains the settings that define the configuration of the management group, the management packs, and the performance, event data and alerts that are currently being collected and processed by agents.

 OperationsManagerDW

This database contains data that has been processed and prepared for reporting. Common objects and attributes are stored in the shared area of the database. Other data, mainly unique data types, is stored in a separate area of the database that is not shared.

Multiple management groups can store data in a single OperationsManagerDW database.

Typically, these two databases synchronize every 30 seconds to ensure that the OperationsManagerDW database is up to date.

Data Warehouse Server

The Data Warehouse Server hosts the OperationsManagerDW database.

Management Server

The management server processes Write actions that are included in management packs. Write actions are instructions for writing performance and event data to the OperationsManagerDB database.

Operations Manager Reporting Server

The Operations Manager Reporting Server builds and presents reports. It runs Internet Information Services (IIS) with the SQL Server Reporting Services (SSRS) Web site and services.

The Operations Manager Reporting Server and the Data Warehouse Server can be deployed on the same server or on separate servers.

The Operations Manager Reporting Server must be located in the same domain as the Operations Manager system or in a trusted domain. Each management group requires a separate instance of SQL Server 2005 with SQL Reporting Services.

Reporting Data Warehouse

The Reporting Data Warehouse manages the OperationsManagerDW database. You can set up this component from Operations Manager Setup.

Reports

Operations Manager Reporting supports the following report types:

 Published reports - These reports are automatically available in the Operations Console after you install Operations Manager and the Reporting feature. These predefined reports are generic, allowing you to easily configure them to meet the needs of your organization.  
The Microsoft Generic Report Library contains some of the predefined reports. You can use these reports with any management pack and with most standard Operations Manager 2007 data types. Unlike Microsoft Operations Manager (MOM) 2005, these generic reports can be used even if the management pack does not include reports.

 Linked reports—These reports are based on existing reports from the Generic Report Library. These reports are defined in custom management packs, and they are available in the Operations console, in the Reporting pane.

 Custom reports— These reports are authored from queries that you build in Microsoft Visual Studio. There are several levels of customization: simple, moderate, and advanced.

 Report solution—These reports are defined with Visual Studio and are available in a management pack. This customization requires the highest level of expertise. This guide does not provide guidance for creating this type of report.

For additional information about reports, including the procedures for creating the different types of reports, see the “Report Types” section.

Report Parameter Area

Operations Manager 2007 extends SQL Server Reporting Services (SSRS) by introducing the report parameter definition area (parameter area). The parameter area provides a user-friendly interface and flexibility for entering parameter values at runtime. You can configure reports with a requirement for runtime parameters that define the scope of the report results.

When you run a report, the parameter area is displayed, enabling you to enter the requested values. Then the report runs and displays the resulting data, which is based on the values that you entered. The parameter area functionality is unique to Operations Manager and is not supported by SQL Reporting Services.

Operations Manager stores the report parameter definitions in a Report Parameter Definition Language (.rpdl) file. You must store this file in the same folder as the report file itself, and you must name it with the same file name as the report, but with an .rpdl extension.

In Operations Manager, all the report parameter controls for the predefined reports are stored in the Microsoft.SystemCenter.DataWarehouse.Report.Library management pack.

Management Packs

Management packs can contain write actions to write collected performance and event data to the OperationsManagerDW database. The management server processes these write actions and writes data to the OperationsManagerDW database according to the instructions in the write action.

When you install the Reporting feature, the following predefined management packs, which contain internal system information about Reporting, are loaded automatically:

 Microsoft.SystemCenter.DataWarehouse.Internal.xml

This management pack contains definitions that are related to the OperationsManagerDW database. This management pack also contains definitions for classes and relationships in the OperationsManagerDW database that the management group needs.

 Microsoft.SystemCenter.DataWarehouse.Ownership.Internal.xml

This management pack contains definitions of classes and relationships in the OperationsManagerDW that the management group needs.

 Microsoft.SystemCenter.DataWarehouse.Report.Library.xml

This management pack contains display strings and custom control definitions for the predefined reports such as the Availability report.

 Microsoft.SystemCenter.DataWarehouse.Library.xml

This management pack contains definitions related to the OperationsManagerDW database that Reporting depends on.

Reporting-Related Write Actions in management packs

To support reports, management packs, such as the SQL Server 2005 management pack, contain the following sections:

 Write action modules that contain instructions for writing data that was collected by an Operations Manager agent to a database.

 Performance data write action—Write actions that contain instructions for writing performance data directly to the OperationsManagerDW database.

The following is an example of a performance data write action:

<WriteActions>

<WriteAction ID="WA" TypeID="DataWarehouse!Microsoft.SystemCenter.DataWarehouse.PublishPerformanceData"/>

</WriteActions>

 Event data write action—Write actions that contain instructions for writing event data directly to the OperationsManagerDW database.

The following is an example of an event data write action:

<WriteActions>

<WriteAction ID="WA" TypeID="DataWarehouse!Microsoft.SystemCenter.DataWarehouse.PublishEventData"/>

</WriteActions>

The following code sample shows how the Base operating system management pack writes a collected event data simultaneously to both the OperationsManager and the OperationsManagerDW databases:

<WriteActions>

<WriteAction ID="WriteToDB" TypeID="SC!Microsoft.SystemCenter.CollectEvent" />

<WriteAction ID="WriteToDW"TypeID="SCDW!Microsoft.SystemCenter.DataWarehouse.PublishEventData"/>

</WriteActions>

What's New

The Reporting feature in Operations Manager 2007 has been redesigned to focus on scenario-based reporting, performance, and usability.

Reporting in Operations Manager 2007 has changed as follows:

 Interface enhancements:

 The Reporting console is now integrated into the Operations Manager 2007 Operations Console.

 When selecting objects, you can now search for specific items.

 Report configuration enhancements:

 You can now configure dates as relative dates. This functionality allows you to configure reports with a recurring schedule that is aligned with the Information Technology (IT) cycles in your organization.

 Embedded native support for business hours is available in every report.

 A newly implemented parameter area makes it easier to customize reports and to find, select, and schedule reports.

 Report data enhancements:

 Reporting data is now available in near real time, with only very short latency or no latency at all.

 Reporting data is now pre-aggregated, summarized, and indexed, making long-term reporting and historical data immediately accessible. Additionally, in the OperationsManagerDW database, data is aggregated and available in different aggregation units.

 Database enhancements:

 Scalability of the OperationsManagerDW database is improved.

 Maintenance of the OperationsManagerDW database is now built in. This automatically improves performance whenever possible.

 The schema in the OperationsManagerDW database is now improved and easier to query on.

 Write action modules can now write data directly to both the OperationalManager database and to the OperationsManagerDW database. You can choose to write data to either database or to both in parallel.

Working with Reports

Operations Manager Reporting supports the following reports:

 Published reports - These reports are automatically available in the Operations console after you install Operations Manager and the Reporting feature. These predefined reports are generic, enabling you to configure them to meet the needs of your organization.  
  
Published reports support the configuration of a report with prepopulated parameters.  
Published reports are created and managed in the console. After you create them, you can access them in the Reporting pane, under Authored Reports.

 Linked reports—These reports are based on existing reports from the Generic Report Library. Linked reports support the configuration of a report with prepopulated parameters and configuration of the visibility of parameters at runtime.

These reports are defined in custom management packs, and they are available in the Operations console, in the Reporting pane.

 Custom reports— These reports are authored from queries that you build in Microsoft Visual Studio. There are several levels of customization: simple, moderate, and advanced.

 Report solution—These reports are defined with Visual Studio and are available in a management pack. This customization requires the highest level of expertise. This guide does not provide guidance for creating this type of report.

The following sections provide additional information about each report type and scenarios to walk you through the process of creating and authoring reports.

Publishing a Generic Report

Published reports are based on existing reports from the Generic Report Library. You can configure a published report with prepopulated parameters and then make it available to other users. You can also share a published report outside of Operations Manager, such as on a SharePoint site.

Publish and Share a Report - Sample Scenario

This section describes the following scenarios:

 How to publish the Availability Report as an Authored Report.

 How to share the report with other users

 How to publish a report to a SharePoint site (or another collaboration tool)

After you have created the report, you can schedule it to run or to publish to other users. For information, see the “Schedule reports” section.

Publishing the Availabiity Report

Procedures

To publish the Availability Report as an Authored Report

|  |
| --- |
| 1. In the Operations console, select Reporting.  2. In the Reporting pane, expand Reporting, and then select Microsoft Generic Report Library.  3. In the Microsoft Generic Report Library Reports pane, double-click the Availability report to open it.  4. In the report window, populate the required parameters.  5. Click Run, and ensure that the report ran successfully.  6. In the report results window, select File, and then select Publish.  7. In the Publish to Authored Reports dialog box, enter a meaningful name and description for the report. Click OK and close all report windows.  8. In the Operations console, in the Reporting pane, you can now select Authored Reports and the newly published report will be listed in the Authored Reports Reports pane.  You can now run this report with no further user intervention. |

Sharing the Availability Report

SQL Server Reporting Services (SSRS) report files are stored under the My Reports folder - reports in that folder are available only to their respective creator. Operations Manager 2007 searches the SRSS My Reports folder for report files. Report files that are found are then displayed in the console, in the Reporting pane.

If you created a report, you can extend the availability of that report to other users by storing the report file in a custom folder within SSRS.

In this sample scenario, initially, the newly published Availability report is stored in the SSRS My Reports folder and is therefore available only to the user who published it. To share that report with other users, you need to move it to a new, custom folder within SSRS that is viewable to all other users.

Procedures

To share the Availability report with other users

|  |
| --- |
| 1. Open Internet Explorer, and open the SQL Server Reporting Services (SSRS) homepage by entering the following URL: http://<servername>|localhost>/reports.  2. On the SQL ServerReporting Services Home page, click New Folder.  3. On the New Folder page, fill in a Name and a Description for the new folder and then click OK.  4. Back on the Home page, select the My Reports folder and on the My Reports page, click your published report to run the report.  5. In the report results page, select Properties.  6. On the General page, click Move and then select the custom folder that you previously created. Click OK. Click Home to go back to the Home page.  7. On the Home page, click the My Reports folder.  On the My Reports page, click Show Details. Select the Edit button that corresponds to the report's .rpdl file. In the file .rpdl page, click Move, and then select the same custom folder that you previously created. Click OK.  8. In the Operations Manager console, select Reporting.  9. In the Reporting pane, expand Reporting, and then select the custom folder that you previously created.  The report is displayed in the results pane, and is now available to all users with appropriate reporting access. |

Publishing the Availability Report to Sharepoint

When you publish the Availability report by following the previous steps, you generate a report with prepopulated parameters. You can view this report by running it through the Report server URL. Because Operations Manager is not required to render the report, you can view the report within Windows SharePoint Services.

Procedures

To publish a report to a SharePoint site

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Run the report viewer through the Report server URL to generate the unique URL for the report.  2. Copy the generated URL for the report.  3. In SharePoint, add either a new Page Viewer Web part or the Report Viewer.  4. If you add a Page Viewer Web part, paste the report URL in the Link field.  5. If you add a Report Viewer, paste the report URL in the Report Manager URL field and type a path for the report (such as NewReports/ThisMonthAvailability).  6. To customize the appearance of the report, edit the URL to show or hide parameters and toolbars. To do this, use the following text string:   |  |  | | --- | --- | | URL text | Purpose | | &rc:Parameters=Collapsed | Collapses the parameter bar in the display but includes the parameters in the URL in the browser.  For example: http://<servername>/Reportserver?%2fNewReports%fThisMonthAvailability&rs:Command=Render&rc:Parameters=Collapsed | | &rc:Parameters=true | Shows the parameter bar in the display. This is the default.  For example: http://<servername>/Reportserver?%2fNewReports%fThisMonthAvailability&rs:Command=Render&rc:Parameters=True | | &rc:Parameters=false | Hides the parameters in the view and does not pass them to the browser.  For example: http://<servername>/Reportserver?%2fNewReports%fThisMonthAvailability&rs:Command=Render&rc:Parameters=false | | &rc:toolbar=false | Hides the toolbar.  For example: http://<servername>/Reportserver?%2fNewReports%fThisMonthAvailability&rs:Command=Render&rc:toolbar=false | |

Using Linked Reports

A linked report is a shortcut to a report—it is similar to a program shortcut on your desktop. A linked report is derived from publicly defined reports from any management pack. A linked report retains some of the original report's properties, such as the report layout. Other properties of the linked report, such as parameters and subscriptions, can be different from the original report.

Operations Manager 2007 takes advantage of linked reports to configure reports with pre-populated default parameter values and with hidden parameters. Configuring reports with pre-populated parameters ensures that the report runs consistently every time you run it. Operations Manager implements linked reports in management packs, such as the SQL Server and the Windows Server management packs.

For more information about SQL Server Reporting Services linked reports see the [Microsoft MSDN SQL Server Developer Center](http://go.microsoft.com/fwlink/?LinkId=106558) Web site. For more information about specific parameter values and parameter controls that can be used when creating linked reports, see Appendix A.

Types of Linked Reports

Linked reports can be created from any of the Generics reports that are included in the “Microsoft.SystemCenter.DataWarehouse.Report.Library” management pack.

The available reports are described in the following table.

|  |  |
| --- | --- |
| Report type | Report name |
| Availability Reports | Microsoft.SystemCenter.DataWarehouse.Report.Availability  Microsoft.SystemCenter.DataWarehouse.Report.AvailabilityTime  Microsoft.SystemCenter.DataWarehouse.Report.AvailabilityMonitor  Microsoft.SystemCenter.DataWarehouse.Report.ConfigurationChange  Microsoft.SystemCenter.DataWarehouse.Report.AvailabilityHealth |
| Alert Logging Latency Report | Microsoft.SystemCenter.DataWarehouse.Report.AlertLoggingLatency |
| Alert Report | Microsoft.SystemCenter.DataWarehouse.Report.Alert |
| Event Analysis Report | Microsoft.SystemCenter.DataWarehouse.Report.EventAnalysis |
| Alert Detail Report | Microsoft.SystemCenter.DataWarehouse.Report.AlertDetail |
| Performance Reports | Microsoft.SystemCenter.DataWarehouse.Report.Performance  Microsoft.SystemCenter.DataWarehouse.Report.PerformanceDetail |
| Performance Top Report | Microsoft.SystemCenter.DataWarehouse.Report.PerformanceTop |
| Performance Top Instance Report | Microsoft.SystemCenter.DataWarehouse.Report.PerformanceTopInstance |
| License Report | Microsoft.SystemCenter.DataWarehouse.Report.License |
| Custom Configuration Report | Microsoft.SystemCenter.DataWarehouse.Report.CustomConfiguration |
| Event Template Report | Microsoft.SystemCenter.DataWarehouse.Report.EventTemplate |
| Most Common Event Report | Microsoft.SystemCenter.DataWarehouse.Report.MostCommonEvents |
| Most Common Alerts Report | Microsoft.SystemCenter.DataWarehouse.Report.MostCommonAlerts  Microsoft.SystemCenter.DataWarehouse.Report.Downtime |

Each of these reports have nodes for parameters definitions called “ReportParameterControl”.

When you create a linked report in a management pack, you actually create a reference to an existing report, with a custom configuration for all of the parameters required by the report.

Creating a Linked Report

Using the XML schema in a custom management pack, you can define linked reports. After you import a management pack, you can access linked reports in the Reporting pane in the console by expanding Reporting and selecting the respective management pack.

The sample linked.reporting.xml file is used as the basis for creating linked reports. Within this file, the <LinkedReports> section in the <Reporting> section contains information about linked reports.

In the <LinkedReports> section, the ID, Accessibility, and Visible attributes are similar to those attributes in the original report. The Base attribute is unique to linked reports, and it describes the report that this linked report is based on. In this case, it is the generic Availability report. The <References> section contains information about the original generic report. The <LanguagePacks> section contains information about the management pack and about the report display name.

In this sample scenario, you create an Availability report with preconfigured settings for data aggregation, time range, and downtime. In this scenario, Availability is defined by the System.State.EntityHealth monitor; the Warning (2) and Unmonitored (3) states are considered as downtime. You configure the report for daily aggregations and calculate data for business hours only.

Procedures

To create a linked report with prepopulated parameters

|  |
| --- |
| 1. Identify the version and the Public Key Token of the Microsoft.SystemCenter.DataWarehouse.Library management pack as follows:  a. In Visual Studio, open the Microsoft.SystemCenter.DataWarehouse.Report.Library.{\*}.xml file in the  %SystemDrive%:\Program Files\System Center Operations Manager 2007\Health Service State\Management Packs\ folder.  b. Press Control + K, and then press Control + D to redisplay the .xml file in a more readable format.  c. Locate the <Identity> section, and note the values of <Version> and <PublicKeyToken>.  2. Open the sample Linked.reporting.xml file in an XML editor such as Visual Studio. Locate the <References> section, and then update the values of the <Version> and <PublicKeyToken> entries with the values that you noted from the Data Warehouse Library management pack above.  3. Update the <Parameters> section of the .xml file with preconfigured parameter values. When you are running the report, it will always be prepopulated with these values.  a. Add an entry to set DataAggregation to daily.  b. Add an entry to set the DownTime parameter.  c. Add an entry to enable business hours reporting only.  d. Add an entry to configure the System.Health.EntityState monitor to calculate availability.  e. Add entries to configure business hours reporting between 8 a.m. and 5 p.m. Monday through Friday only and to configure the default report range from Friday of the previous week to today.  After the additions, the .xml file should contain the following:  <Parameters>  <Parameter Name="DataAggregation">  <Value>1</Value>  </Parameter>  <Parameter Name="DownTime">  <Value>2</Value><Value>3</Value>  </Parameter>  <Parameter Name="TimeType">  <Value>Business</Value>  </Parameter>  <Parameter Name="MonitorName">  <Value>System.Health.EntityState</Value>  </Parameter>  <Parameter Name="StartDate\_BaseType">  <Value>Friday</Value>  </Parameter>  <Parameter Name="StartDate\_OffsetType">  <Value>Week</Value>  </Parameter>  <Parameter Name="StartDate\_OffsetValue">  <Value>-1</Value>  </Parameter>  <Parameter Name="EndDate\_BaseType">  <Value>Today</Value>  </Parameter>  <Parameter Name="EndDate\_OffsetType">  <Value>None</Value>  </Parameter>  <Parameter Name="EndDate\_OffsetValue">  <Value>0</Value>  </Parameter>  <Parameter Name="TimeType">  <Value>Business</Value>  </Parameter>  <Parameter Name="StartDate\_BaseValue">  <Value>10/26/2007 08:00:00 AM</Value>  </Parameter>  <Parameter Name="EndDate\_BaseValue">  <Value>10/26/2007 5:00:00 PM</Value>  </Parameter>  <Parameter Name="TimeWeekMap">  <Value>Monday</Value>  <Value>Tuesday</Value>  <Value>Wednesday</Value>  <Value>Thursday</Value>  <Value>Friday</Value>  </Parameter>  </Parameters>  Note  The StartDate\_BaseValue and EndDate\_BaseValue strings must include a valid date and time. However, only the time string is used when running the report.  4. Save the updated .xml file.  5. In the Operations Manager Operations Console, import the customized report as follows:  a. Select Administration.  b. In the Administration pane, right-click Administration and then select Import Management Packs.  c. In the Select Management Packs to import dialog box, select the Linked.reporting.xml file and click Open.  d. Select Reporting. In the Reporting pane, expand Reporting and select Linked Reports. The ApplicationAvailability Linked Report is now listed in the Linked Reports Reports pane.  6. Right-click the report and select Open to run the report. The parameters are now pre-populated as defined in the custom management pack .xml file. |

Hide the Parameter Area

Because these parameters are now pre-populated and cannot be changed at runtime, you can customize the .xml file so that they are not visible when running the report.

In this sample scenario, you will customize the .xml file to not display the DownTime and DataAggregation parameters when the report runs.

To configure a linked report with hidden parameters

|  |
| --- |
| 1. Open the custom .xml file in an XML editor such as Visual Studio.  2. Locate the <ParameterBlock> section in the file, and replace the entire <Controls> section with the following code:  <Controls>  <Control type=  "Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectXmlPicker" columnSpan="2" rowSpan="2">  <ReportParameters>  <ReportParameter name="ObjectList">  <Prompt> Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ObjectList  </Prompt>  </ReportParameter>  <ReportParameter name="ManagementGroupId" binding="GroupList"/>  </ReportParameters>  </Control>  <Control type=  "Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.BusinessRelativeDateTimePicker">  <ReportParameters>  <ReportParameter name="TimeZone" binding="TimeZone">  <Prompt>  Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.TimeZone  </Prompt>  </ReportParameter>  <ReportParameter name="TimeZoneName" binding="TimeZoneName" />  <ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />  <ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue">  <Prompt>  Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>  </ReportParameter>  <ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />  <ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />  <ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />  <ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">  <Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>  </ReportParameter>  <ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />  <ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />  <ReportParameter name="TimeType" binding="TimeType" />  <ReportParameter name="TimeWeekMap" binding="TimeWeekMap" />  </ReportParameters>  </Control>  </Controls>  3. Save the updated .xml file.  4. In the Operations console, import the customized report as follows:  a. Select Administration.  b. In the Administration pane, right-click Administration and then select Import Management Packs.  c. In the Select Management Packs to import dialog box, select the Linked.reporting.xml file and click Open.  d. In the console, select Reporting. In the Reporting pane, expand Reporting and then select Linked Reports.  e. In the Linked Reports Reports pane, right-click Availability Linked Report and then select Open to run the report. The pre-populated parameters are now displayed.  5. Add an object and run the report. The report runs according to the specified parameters in the custom .xml file. |

Authoring Custom Reports

With Operations Manager 2007, you can author custom reports by using Visual Studio. You can then access these reports from the Operations Manager console in the Reporting pane.

Reporting supports the following three levels of complexity when customizing reports:

 Simple—A standard SQL Server Reporting Services (SSRS) report.

 Moderate—A standard SSRS report that is managed in a management pack.

 Advanced—An SSRS report with smart parameter controls that is managed in a management pack.

In general, to author custom reports using this guide, you need a good understanding of the following:

 SQL Server Reporting Services.

 The schema that you are selecting from. A basic discussion of the Reporting Management Pack schema is provided in Appendix A. For detailed information, including graphs, see the [Database Reference - OperationsManagerDW](http://go.microsoft.com/fwlink/?LinkId=106561) Web page.

 How Operations Manager 2007 Reporting works.

 How to build a Smart Parameter Header.

For information about specific data types and sample queries for operational, performance, and discovery data that can be used when creating custom reports, see Appendix B.

Report Authoring pre-requisites

You can author reports on any computer that meets the following criteria:

 Visual Studio .NET 2005 installed.

 SQL Server Reporting Client Components installed.

 The custom extension .dll libraries installed For more information about enabling the EnterpriseManagementChartControl, see <http://go.microsoft.com/fwlink/?LinkId=111034>

 A connection to the System Center data warehouse and SQL Reporting Services 2005 SP1 servers, which will be used for report development.

 The SQL Reporting Services Server with a Dundas Chart control version > 1.5 installed.

The first step in creating a report in Report Designer is to create a report project. A report project acts as a container for report definitions and resources. Every file in the report project is published to the report server when the project is deployed. When you create a project for the first time, a solution is also created as a container for the project. You can add multiple projects to a single solution. This is useful when you want to organize reports into multiple folders since each project corresponds to a single report folder.

You must configure each report project with the URL of the report server and the target folder where the reports in that project should be deployed. In Report Designer, this is done by right-clicking the project name, clicking Properties, and then editing the properties in the Properties dialog box.

A project can contain multiple reports and data sources. Each report can contain multiple data sets (queries) and layout elements (text boxes, charts, tables, lines, etc.).

Custom Reports - Sample Scenario for a Simple Report 1

This first scenario walks through the basics of creating a report using Visual Studio tools. This report collects all Events with an ID of 21025.

Before you start this scenario, create a new Report Server Project type in Visual Studio and then add a new report. After you add a new report, you use the wizard or build it from scratch.

Procedures

To create a report using Visual Studio

|  |
| --- |
| 1. Create the data source, and then click Next.  2. Define the query, and then click Next. In this part you can use the query builder, or use SQL 2005 query tools.  Use the following query:  SELECT evt.eventnumber,Evt.EventOriginId,  Evt.EventPublisherRowId,  Pub.EventPublisherName,  Evt.DateTime,  Evt.EventChannelRowId,  Chl.EventChannelTitle,  Evt.EventCategoryRowId,  Chl.EventChannelTitle,  Evt.EventLevelId,  Lev.EventLevelTitle,  Evt.LoggingComputerRowId,  Cmp.ComputerName,  Evt.EventDisplayNumber,  Evt.UserNameRowId,  Usr.UserName,  Det.RawDescription,  Det.RenderedDescription,  Det.EventData  FROM Event.vEvent(NoLock) Evt  Inner Join EventPublisher(NoLock) Pub  On Evt.EventPublisherRowId = Pub.EventPublisherRowId  Inner Join EventChannel(NoLock) Chl  On Evt.EventChannelRowId = Chl.EventChannelRowId  Inner Join EventCategory(NoLock) Cat  On Evt.EventCategoryRowId = Cat.EventCategoryRowId  Inner Join EventLevel(NoLock) Lev  On Evt.EventLevelId = Lev.EventLevelId  Inner Join EventLoggingComputer(NoLock) Cmp  On Evt.LoggingComputerRowId = Cmp.EventLoggingComputerRowId  Inner Join EventUserName(NoLock) Usr  On Evt.UserNameRowId= Usr.EventUserNameRowId  Inner Join Event.vEventDetail (NoLock) Det  On Evt.EventOriginId = Det.EventOriginId  WHERE Evt.EventNumber = 21025  3. Select the report type, Matrix or Table, and then click Next.  4. Select the fields to be displayed on the different areas of the report, and then click Next.  5. Select the table layout, Stepped or Block, and then click Next.  6. Select the style of the layout. You have the following style choices:   Slate   Forest   Corporate   Bold   Ocean   Generic  A preview of each style is displayed in the right pane of the window when you select the style.  Click Next.  7. You are prompted for the report name. This window also summarizes the information for the report.  8. Click Finish. You can now see and modify the report.  9. To incorporate the definition (xml) into the management pack, select the report in the project explorer, right-click the report, and then click View Code.  10. Copy the content.  <Reporting>  <Reports>  <Report ID="Demo.MyReport" Accessibility="Public" Visible="true">  <Definition>  [Report Definition]  </Definition>  </Report>  </Reports>  </Reporting>  Note  Be sure to omit the xml declaration before pasting it in the management pack Reporting section: <?xml version="1.0" encoding="utf-8"?> |

Custom Reports - Sample Scenario for a Simple Report 2

In this scenario, you create a simple custom report that displays a specified performance counter for all managed entities of a particular group. This example uses the SQL 2005 Computer group and the performance counter SQL Server:General Statistics – User Connections.

This example highlights how the DataWarehouse stores the managed entity hierarchy to better understand how to target reports at the correct objects.

Before you start this scenario, complete the following tasks:

 Ensure that the Microsoft SQL Server 2000/2005 Management Pack is installed and that computers that are running SQL Server 2005-based servers are being monitored.

 Choose the dataset that you want to report on. You can use SQL Management Studio to do that.

Procedures

To author a custom report in SQL Management Studio

|  |
| --- |
| 1. Open SQL Management Studio, and connect to the OperationsManagerDW database.  2. Run the following query:  select \* from vmanagedentity  where fullname = Microsoft.SQLServer.2005.ComputerGroup  3. Run the following query to find all the managed entities that are contained by ManagedEntityRowId=x (which was returned by the query):  EXEC [dbo].[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectListParse]'2007-10-26 10:50:23.150','2007-10-26 10:50:23.150', '<Data><Objects><Object Use="Containment">x</Object></Objects></Data>'  Where:  @StartDate is the group containment start date (first parameter above)  @EndDate is the group containment end date (second parameter above)  @ObjectList is the XML with containment object (the xml fragment above)  4. The results display the IDs of all the managed entities that are contained in the Microsoft.SQLServer.2005.ComputerGroup group. You can now add this information to a temporary table and join it to the appropriate performance tables to produce the necessary information.  DROP TABLE #ManagedEntity  CREATE TABLE #ManagedEntity (ManagedEntityRowId int)  INSERT INTO #ManagedEntity  EXEC [dbo].[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectListParse]'2007-10-29 10:50:23.150','2007-10-29 10:50:23.150', '<Data><Objects><Object Use="Containment">45</Object> </Objects></Data>'  --SELECT ManagedEntityRowID FROM tmp\_ManagedEntity  --GO  SELECT  vPerf.DateTime,  vPerf.SampleCount,  vPerf.AverageValue,  vPerf.MinValue,  vPerf.MaxValue,  vPerf.StandardDeviation,  vPerformanceRuleInstance.InstanceName,  vManagedEntity.Path,  vPerformanceRule.ObjectName,  vPerformanceRule.CounterName  FROM Perf.vPerfHourly AS vPerf INNER JOIN  vPerformanceRuleInstance ON  vPerformanceRuleInstance.PerformanceRuleInstanceRowId =  vPerf.PerformanceRuleInstanceRowId INNER JOIN  vManagedEntity ON vPerf.ManagedEntityRowId =  vManagedEntity.ManagedEntityRowId INNER JOIN  vPerformanceRule ON vPerformanceRuleInstance.RuleRowId =  vPerformanceRule.RuleRowId INNER JOIN  #ManagedEntity ON vManagedEntity.ManagedEntityRowId =  #ManagedEntity.ManagedEntityRowId  The following are optional Where clauses for determining performance objects, counters, or instances:  WHERE  (vPerformanceRule.ObjectName IN ('SQLSERVER:General Statistics'))  AND  (vPerformanceRule.CounterName IN ('User Connections'))  ORDER BY  vPerf.DateTime  5. Use Visual Studio to design your report. For more information about using Visual Studio to design reports, see the [Microsoft MSDN, SQL Server Developer Center](http://go.microsoft.com/fwlink/?LinkId=106560) Web page.  6. Add the designed report directly to SSRS. The report is now accessible in the Operations Manager console when you select Reporting.  7. You can include this report in a management pack by adding the report definition code below to the management pack file. Later, when you import that management pack, you will be able to access the report in the console.  <Reporting>  <Reports>  <Report ID="Demo.MyReport" Accessibility="Public" Visible="true">  <Definition>  [Report Definition]  </Definition>  </Report>  </Reports>  </Reporting>  Note  Be sure to omit the xml declaration before past it on the management pack Reporting section.  <?xml version="1.0" encoding="utf-8"?> |

Custom Reports - Sample Scenario for an Advanced Report

This scenario takes the report in the previous section, turns the query into parameters, and adds smart parameter controls to it.

There are three steps for this scenario:

1. Create a stored procedure based on the query from the previous scenario.

2. Create a report based on the stored procedure.

3. Add the report to the management pack.

Creating a stored procedure

The query from the previous scenario can be turned into a stored procedure with the following required parameters:

 StartDate

 EndDate

 ObjectList

You can create this procedure in OperationsManagerDW for designing your report and testing. However, in a production environment, we recommend installing the stored procedure as part of the management pack import process, as described at the end of this section. The stored procedure should be strongly named and unique.

To create the stored procedure

|  |
| --- |
|  Run the following script:  CREATE PROCEDURE Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet  @StartDate\_BaseValue datetime, @EndDate\_BaseValue datetime, @ObjectList xml  AS  DECLARE @ExecError int CREATE TABLE #ManagedEntity (ManagedEntityRowId int)  INSERT INTO #ManagedEntity EXECUTE @ExecError = [Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectListParse] @ObjectList = @ObjectList, @StartDate = @StartDate\_BaseValue , @EndDate = @EndDate\_BaseValue  SELECT  vPerf.DateTime, vPerf.SampleCount,  vPerf.AverageValue, vPerf.MinValue, vPerf.MaxValue, vPerf.StandardDeviation,  vPerformanceRuleInstance.InstanceName,  vManagedEntity.Path,  vPerformanceRule.ObjectName,   vPerformanceRule.CounterName  FROM Perf.vPerfHourly AS vPerf INNER JOIN  vPerformanceRuleInstance ON vPerformanceRuleInstance.PerformanceRuleInstanceRowId =  vPerf.PerformanceRuleInstanceRowId INNER JOIN  vManagedEntity ON vPerf.ManagedEntityRowId = vManagedEntity.ManagedEntityRowId INNER JOIN  vPerformanceRule ON vPerformanceRuleInstance.RuleRowId = vPerformanceRule.RuleRowId INNER JOIN  #ManagedEntity ON vManagedEntity.ManagedEntityRowId = #ManagedEntity.ManagedEntityRowId WHERE vPerf.DateTime >= @StartDate\_BaseValue  and vPerf.DateTime < @EndDate\_BaseValue  and (vPerformanceRule.ObjectName IN ('SQLSERVER:General Statistics')) AND (vPerformanceRule.CounterName IN ('User Connections')) ORDER BY vPerf.DateTime  DROP TABLE #ManagedEntity  GO |

Authoring a report based on a stored procedure

Using Visual Studio, you can design the report using the stored procedure “Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet” as the datasource.

For more information about using Visual Studio to design reports, see the [Microsoft MSDN, SQL Server Developer Center](http://go.microsoft.com/fwlink/?LinkId=106560) Web page.

To author a report from a stored procedure

|  |
| --- |
| 1. As part of the design process the parameters will have to be configured. In the dataset the stored procedure parameters map to the report parameters.  2. Configure the parameters as follows:        Because the data warehouse can support data from multiple management groups, the Object List picker is passed the management group values.  3. For the custom report to work, add a dataset that selects the installed management groups.    4. This is configured as follows:    5. Add the new dataset as a report parameter in the custom report.  Note  The naming of the parameters is crucial as it is controlled by the bindings allowed in the parameter block controls, as shown in Appendix B. |

Adding the report to the management pack

Now that the report is designed and the parameters are updated, you can update your management pack XML to add the authored report.

To add the custom report to the management pack

|  |
| --- |
| 1. Add the RDL definition to the XML file, as follows:  <Reporting>  <Reports>  <Report ID="Demo.MyReport" Accessibility="Public" Visible="true">  <Definition>  [Paste content of the RDL file here...]  </Definition>  </Report>  </Reports>  </Reporting>  2. Add the parameter block information. The section should look like the following:  <ParameterBlock columns="4" xmlns="http://schemas.microsoft.com/mom/reporting/2007/ReportParameterSettings">  <Controls>  <Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectXmlPicker" columnSpan="2" rowSpan="2">  <ReportParameters>  <ReportParameter name="ObjectList">                                       <Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ObjectList</Prompt>  </ReportParameter>  <ReportParameter name="ManagementGroupId" binding="GroupList" />  </ReportParameters>  </Control>  <Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DatePicker">  <ReportParameters>  ReportParameter name="StartDate\_BaseValue">  </ReportParameter>  </ReportParameters>  </Control>  <Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DatePicker">  <ReportParameters>  <ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">  </ReportParameter>  </ReportParameters>  </Control>  </Controls>  </ParameterBlock>  3. Save the file. |

Remember that you can include a stored procedure in your management pack, instead of creating it manually as in this scenario. To install the stored procedure from within the management pack, you need to add the stored procedure to the data warehouse scripts, as shown below. You declare three options when adding a script from a management pack:

 Install - What to do with the script installing the management pack

 Uninstall - What to do when uninstalling the management pack

 Upgrade - What to do during a management pack upgrade

Your management pack is updated to look like the following:

<Reporting>

<DataWarehouseScripts>

<DataWarehouseScript ID="Microsoft.SystemCenter.DataWarehouse.Report.Authoring.PerfDataGet" Accessibility="Public">

<Install><![CDATA[IF NOT EXISTS (SELECT \* FROM sysobjects WHERE type = 'P' AND name = 'Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet')

BEGIN

EXECUTE ('CREATE PROCEDURE dbo.[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet] AS RETURN 1')

END

GO

ALTER PROCEDURE [dbo].[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet]

@StartDate\_BaseValue datetime,

@EndDate\_BaseValue datetime,

@ObjectList xml

AS

DECLARE @ExecError int

CREATE TABLE #ManagedEntity (ManagedEntityRowId int)

INSERT INTO #ManagedEntity

EXECUTE @ExecError = [Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectListParse]

@ObjectList = @ObjectList,

@StartDate = @StartDate\_BaseValue ,

@EndDate = @EndDate\_BaseValue

SELECT

vPerf.DateTime,

vPerf.SampleCount,

vPerf.AverageValue,

vPerf.MinValue,

vPerf.MaxValue,

vPerf.StandardDeviation,

vPerformanceRuleInstance.InstanceName,

vManagedEntity.Path,

vPerformanceRule.ObjectName,

vPerformanceRule.CounterName

FROM Perf.vPerfHourly AS vPerf INNER JOIN

vPerformanceRuleInstance ON vPerformanceRuleInstance.PerformanceRuleInstanceRowId = vPerf.PerformanceRuleInstanceRowId INNER JOIN

vManagedEntity ON vPerf.ManagedEntityRowId = vManagedEntity.ManagedEntityRowId INNER JOIN

vPerformanceRule ON vPerformanceRuleInstance.RuleRowId = vPerformanceRule.RuleRowId INNER JOIN

#ManagedEntity ON vManagedEntity.ManagedEntityRowId = #ManagedEntity.ManagedEntityRowId

WHERE vPerf.DateTime >= @StartDate\_BaseValue

and vPerf.DateTime < @EndDate\_BaseValue

and (vPerformanceRule.ObjectName IN ('SQLSERVER:General Statistics'))

AND (vPerformanceRule.CounterName IN ('User Connections'))

ORDER BY vPerf.DateTime

DROP TABLE #ManagedEntity

GO]]></Install>

<Uninstall><![CDATA[IF EXISTS (SELECT \* FROM sysobjects WHERE type = 'P' AND name = 'Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet')

BEGIN

DROP PROCEDURE dbo.[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet]

END

GO]]></Uninstall>

<Upgrade><![CDATA[IF NOT EXISTS (SELECT \* FROM sysobjects WHERE type = 'P' AND name = 'Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet')

BEGIN

EXECUTE ('CREATE PROCEDURE dbo.[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet] AS RETURN 1')

END

GO

ALTER PROCEDURE [dbo].[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Authoring\_Perf\_DataGet]

@StartDate\_BaseValue datetime,

@EndDate\_BaseValue datetime,

@ObjectList xml

AS

DECLARE @ExecError int

CREATE TABLE #ManagedEntity (ManagedEntityRowId int)

INSERT INTO #ManagedEntity

EXECUTE @ExecError = [Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectListParse]

@ObjectList = @ObjectList,

@StartDate = @StartDate\_BaseValue ,

@EndDate = @EndDate\_BaseValue

SELECT

vPerf.DateTime,

vPerf.SampleCount,

vPerf.AverageValue,

vPerf.MinValue,

vPerf.MaxValue,

vPerf.StandardDeviation,

vPerformanceRuleInstance.InstanceName,

vManagedEntity.Path,

vPerformanceRule.ObjectName,

vPerformanceRule.CounterName

FROM Perf.vPerfHourly AS vPerf INNER JOIN

vPerformanceRuleInstance ON vPerformanceRuleInstance.PerformanceRuleInstanceRowId = vPerf.PerformanceRuleInstanceRowId INNER JOIN

vManagedEntity ON vPerf.ManagedEntityRowId = vManagedEntity.ManagedEntityRowId INNER JOIN

vPerformanceRule ON vPerformanceRuleInstance.RuleRowId = vPerformanceRule.RuleRowId INNER JOIN

#ManagedEntity ON vManagedEntity.ManagedEntityRowId = #ManagedEntity.ManagedEntityRowId

WHERE vPerf.DateTime >= @StartDate\_BaseValue

and vPerf.DateTime < @EndDate\_BaseValue

and (vPerformanceRule.ObjectName IN ('SQLSERVER:General Statistics'))

AND (vPerformanceRule.CounterName IN ('User Connections'))

ORDER BY vPerf.DateTime

DROP TABLE #ManagedEntity

GO]]></Upgrade>

</DataWarehouseScript>  
</DataWarehouseScripts>  
</Reporting>

When you import the management pack, the stored procedure is deployed to the Data Warehouse server.

You can now add a dependency on the stored procedure to the report by updating the following in the XML file:

<Reporting>

<Reports>

<Report ID="Demo.MyReport" Accessibility="Public" Visible="true">

<Dependencies>

<DataWarehouseScript>

Microsoft.SystemCenter.DataWarehouse.Report.Authoring.PerfDataGet

</DataWarehouseScript>

<Definition>

[Paste content of the RDL file here...]

</Definition>

</Report>

</Reports>

</Reporting>

Scheduling and Sending Reports

The procedures in this section enable you to create a schedule to run reports, send reports by e-mail, save reports to a file share, or store reports in the SQL Report Server cache.

Creating a Report Schedule

You can use the following procedure to create a schedule to run a report. However, before you do so, make sure that you create an Availability report and save it as a favorite.

Procedures

To schedule a report to run

|  |
| --- |
| 1. In the Operations console, click Reporting.  2. In the Reporting pane, click Favorite Reports.  3. In the Favorite Reports pane, right-click the Availability report, and then click Schedule.  4. In the Subscribe to a Report Wizard, on the Delivery Settings page, do the following:  a. Type a description in the Description text box.  b. In the Delivery method list, click Report Server File Share.  c. Type a file name for the report in the File name text box.  d. Type a file path for the report in the Path text box.  Note  Because report scheduling supports the Universal Naming Convention (UNC), file names must not end in a backslash.  e. In the Render Format list, click the file format that you want for the report.  f. Type a user name in the Username text box, and then type a password in the Password text box.  g. Click the down arrow in the Writemode list, select the Write mode you want for subsequent files, and then click Next.  Note  The credentials that you entered in step 4g must have the same Write user rights on the file share that you entered in step 4d.  5. In the Subscribe to a Report Wizard, on the Subscription Schedule page, do the following:  a. Select one of the Generate the report options.  b. In TheSubscriptioniseffectivebeginning list, type a start date and a start time for the reports that you want to generate. You can also enter the date when this subscription will end in The subscription expires on list, and then click Next.  6. In the Subscribe to a Report wizard, on the Parameters page, specify a span of time for the report in the From and To lists.  7. Make any other changes you need for this report, and then click Finish. |

Sending a Report Through E-mail

You can configure Reporting to automatically send a published report from Operations Manager to another person through e-mail.

Before you can schedule a report for e-mail delivery, you must configure the e-mail settings in the Report Server by using the Reporting Server Configuration Manager.

Procedures

To configure e-mail settings in the SQL Server 2005 Report Server

|  |
| --- |
| 1. Open the Reporting Services Configuration by clicking Start on the Windows taskbar, point to Programs, point to MicrosoftSQLServer 2005, point to ConfigurationTools, and then click ReportingServicesConfiguration.  2. In Reporting Services Configuration Manager, in the Report Server Installation Instance dialog box, click Connect.  3. In the navigation pane, click Email Settings.  4. In the E-mail Settings pane, enter the following:   An e-mail address that will be used as the sender address in the Sender Address text box.   The address of the SMTP server that will be used to send the e-mail messages in the SMTP Server text box.  5. Click Apply, and then click Exit. |

To e-mail scheduled reports

|  |
| --- |
| 1. In the Operations console, click the Reporting button.  2. In the Reporting pane, click Favorite ReportsReports.  3. In the Favorite Reports pane, right-click the Availability report, and then click Schedule.  4. In the Subscribe to a Report Wizard, on the DeliverySettings page, do the following:   Type a description in the Description text box.   In the Delivery method list, click Report Server E-Mail.   Type an e-mail address of the destination inbox to receive reports in the To text box. You can also type e-mail addresses in the Cc, Bcc, and the Reply To text boxes.   In the Render Format list, click the file format you want for the report.   In the Priority list, select the appropriate priority.   Type a subject for the e-mail in the Subject text box.   Click Next.  5. On the Subscription Schedule page, do the following:   Select one of the Generate the report options.   In the The Subscription is effective beginning list, type a start date and a start time for the reports that you want to generate. You can also enter the date when this subscription will end in The subscription expires on list, and then click Next.  6. On the Parameters page, specify a span of time for the report in the From and To lists, make any other changes that you need for this report, and then click Finish. |

Delivering a report to the SQL Server Report Server Cache

You can create a schedule for sending reports to the cache in the SQL Server 2005 Report Server. When you do this, you shorten the time required to retrieve a report, which can be helpful if a report is large or is accessed frequently. For more information about report caching, see <http://go.microsoft.com/fwlink/?LinkId=77536>.

Procedures

To schedule the delivery of a report to the SQL Report Server Cache

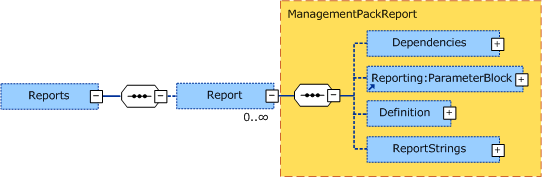
|  |
| --- |
| 1. In the Operations console, click Reporting.  2. In the Reporting pane, click Favorite Reports.  3. In the Favorite Reports pane, right-click the availability report you saved as a favorite, and then click Schedule.  4. In the Subscribe to a Report Wizard, on the Delivery Settings page, do the following:  a. Type a description in the Description text box.  b. In the Delivery method list, click Null Delivery Provider.  c. Click Next.  5. On the Subscription Schedule page, do the following:  a. Select one of the Generate the report options.  b. In the The Subscription is effective beginning list, type a start date and a start time for the reports that you want to generate. You can also enter the date when this subscription will end in The subscription will end list, and then click Next.  6. On the Parameters page, specify a span of time for the report in the From and To lists, make any other changes you need for this report, and then click Finish. |

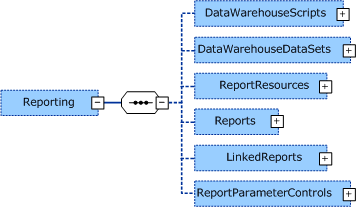
Appendix A - Reporting Management Pack Schema

Within a management pack such as the Microsoft.SystemCenter.DataWarehouse.Report.Library.xml, the Reporting section contains the necessary information for deploying related items such as the reports themselves, linked reports, and report parameter controls. When creating a reporting solution, it is important to understand the management pack schema. The full management pack schema is available at <http://go.microsoft.com/fwlink/?LinkId=111035>.

The following sections describe the basic elements of the schema.

Reports section:





Dependencies: Details any dependencies such as stored procedures required for the report to function.

Parameter Block: Defines the look of the Smart Parameter Header (SPH) that shows up in the console when the report is opened.

Each control in the SPH is bound to one or more ReportParameters

Also, some controls have user-definable properties, such as the LinkedPerformanceChartObjectPicker. All parameter block controls are defined in Appendix B.

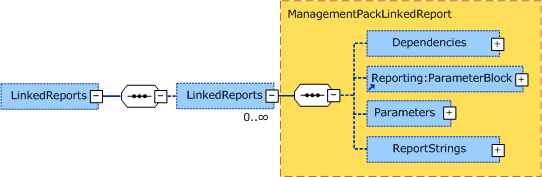
Definition: Contains the report RDL information. When you have an RDL file ready, the XML is copied directly into this section.

Note

Be sure to omit the xml declaration before pasting it in the definition section: <?xml version="1.0" encoding="utf-8"?>

Report Strings: Enables localization for the report.

Linked reports section:



Dependencies: Details any dependencies such as stored procedures required for the report to function.

Parameter Block: Defines the look of the Smart Parameter Header that shows up in the console when the report is opened.

Each Control in the SPH is bound to one or more ReportParameters

Also, some controls have user-definable properties, such as the LinkedPerformanceChartObjectPicker. All parameter block controls are defined in Appendix B.

Parameters: Passes pre-defined parameters to the report to enable the scoping of a generic report to user-defined values.

Report Strings: Enables localization for the report.

Report Parameter Controls section:

The Report parameters block is the place where all these parameters are entered. The Report parameters block consists of a set of report parameter controls, each of which defines a visual representation for one or more underlying report parameters.

Report parameter controls used for a particular report are defined in the RPDL file.

Here is the schema for them:

<xs:element name="Control" minOccurs="1" maxOccurs="unbounded">  
    <xs:complexType>  
        <xs:sequence>

            <xs:element name="ReportParameters" minOccurs="0" maxOccurs="1">  
                <xs:complexType>  
                    <xs:sequence>  
                        <xs:element name="ReportParameter"   
                                           minOccurs="1" maxOccurs="unbounded">  
                            <xs:complexType>  
                                <xs:sequence>  
                                    <xs:element name="Prompt" type="xs:string"  
                                                       minOccurs="0" maxOccurs="1" />  
                                </xs:sequence>  
                                <xs:attribute name="name" type="xs:token" use="required" />  
                                <xs:attribute name="binding" type="xs:token" use="optional" />  
                            </xs:complexType>  
                        </xs:element>  
                    </xs:sequence>  
                </xs:complexType>  
            </xs:element>

            <xs:element name="Properties" minOccurs="0" maxOccurs="1">  
                <xs:complexType>  
                    <xs:sequence>  
                        <xs:element name="Property"  
                                           minOccurs="1" maxOccurs="unbounded">  
                            <xs:complexType>  
                                <xs:sequence>  
                                    <xs:element name="Value" type="xs:string"  
                                                       minOccurs="1" maxOccurs="1" />  
                                </xs:sequence>  
                                <xs:attribute name="name" type="xs:token" use="required" />  
                            </xs:complexType>  
                        </xs:element>  
                    </xs:sequence>  
                </xs:complexType>  
            </xs:element>

        </xs:sequence>  
        <xs:attribute name="type" type="xs:token" use="optional" />  
        <xs:attribute name="columnSpan" type="xs:unsignedByte" use="optional" />  
        <xs:attribute name="rowSpan" type="xs:unsignedByte" use="optional" />  
    </xs:complexType>  
</xs:element>

As you can see from the schema, the parameter control has a list of underlying report parameters it uses. The way it uses these parameters is defined by the binding attribute. Binding is defined by specific control code. You cannot change the way it is handled, but you can set which report parameter is used. "Empty" or default binding is usually used for a report parameter for which the control sets value, but this is not a strict rule.

The next section is the parameter control properties. It is a name multi-value collection. These properties changes control behavior or visual appearance, for example controlling the maximum width or the background color.

All "out of the box" report parameter controls are defined in the Microsoft.SystemCenter.DataWarehouse.Report.Library management pack. The most commonly used controls are defined in Appendix B.

Appendix B - Parameter Values and Parameter Controls

This appendix provides information about specific parameter values and parameter controls that can be used when creating linked reports.

Parameter Values

DateTime

Parameters: StartDate\_BaseType,EndDate\_BaseType

BaseType Values:

 Fixed

 Today

 Monday

 Tuesday

 Wednesday

 Thursday

 Friday

 Saturday

 Sunday

 FirstDayMonth

 LastDayMonth

 FirstDayQuarter

 LastDayQuarter

 FirstDayYear

 LastDayYear

Parameters: StartDate\_OffsetType, EndDate\_OffsetType

OffsetType Values:

 None

 Day

 Week

 Month

 Quarter

 Year

Example:

<Parameter Name="StartDate\_BaseType">

<Value>Friday</Value>

</Parameter>

<Parameter Name="StartDate\_OffsetType">

<Value>Week</Value>

</Parameter>

<Parameter Name="StartDate\_OffsetValue">

<Value>-1</Value>

</Parameter>

<Parameter Name="EndDate\_BaseType">

<Value>Today</Value>

</Parameter>

<Parameter Name="EndDate\_OffsetType">

<Value>None</Value>

</Parameter>

<Parameter Name="EndDate\_OffsetValue">

<Value>0</Value>

</Parameter>

Aggregation Parameter

|  |  |
| --- | --- |
| Daily | 1 |
| Hourly | 0 |

Downtime Parameter

|  |  |
| --- | --- |
| Warning | 2 |
| Planned Maintenance | 6 |
| Unplanned Maintenance | 5 |
| Monitor disabled | 4 |
| Unmonitored | 3 |

Threshold

|  |  |
| --- | --- |
| 1 second(s) | 1 |
| 2 second(s) | 2 |
| 5 second(s) | 5 |
| 10 second(s) | 10 |
| 15 second(s) | 15 |
| 30 second(s) | 30 |
| 1 minute(s) | 60 |
| 2 minute(s) | 120 |
| 5 minute(s) | 300 |
| 10 minute(s) | 600 |
| 15 minute(s) | 900 |
| 30 minute(s) | 1800 |

Chart Scale

|  |  |
| --- | --- |
| Hourly | 0 |
| Daily | 1 |
| Monthly | 2 |

Severity

|  |  |
| --- | --- |
| Information | 0 |
| Warning | 1 |
| Critical | 2 |

Priority

|  |  |
| --- | --- |
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |

Criteria

|  |  |
| --- | --- |
| Defined In | 0 |
| Applied To | 1 |

Exclude Sealed MP

Boolean

Histogram

|  |  |
| --- | --- |
| No | 0 |
| Daily | 1 |
| Weekly | 2 |
| Monthly | 3 |

3D Chart

Boolean

Objects

Example:

<Data>  
 <Values Title="Test">  
 <Value>  
 <Object Use="Containment">9</Object>  
 <Rule>7c512fe8-a3c4-1049-9891-88b259cb5f47</Rule>

<Color>63,63,255</Color>

<Type>Line</Type>

<Scale>1</Scale>

</Value>

<Value>

<Object Use="Self">9</Object>

<Rule>406fc269-8b85-b13a-9079-86dd1df5831b</Rule>

<Color>0,159,0</Color>

<Type>Line</Type>

<Scale>1</Scale>

</Value>

</Values>

</Data>

Where

<Object>–Object Row Id

<Rule>–Rule GUID

<Color>–In the format of R,G,B.

<Type>–One of: Area, Column, Line, Point, Spline, SplineArea, or StepLine.

<Scale>–Double

Report Parameter Controls

The most commonly used report parameter controls in custom reports are as follows:

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.CheckedListBox

Description: A way to select multiple values from a list of valid ones. Replaces standard SSRS multi-value value picker.

Properties:

Bindings

Default (report parameter has to be a multi-value parameter with a list of valid values)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.CheckedListBox" rowSpan="1" columnSpan="1">

<ReportParameters>

<ReportParameter name="Severity"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

<Parameters>

<Parameter Name="Severity">

<Value>0,1,2</Value>

</Parameter>

</Parameters>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.ComboBox

Description: A way to select a value from a set. Replaces standard SSRS value picker control.

Properties:

Bindings

Default (report parameter has to have a list of valid values)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.ComboBox" rowSpan="1" columnSpan="1">

<ReportParameters>

<ReportParameter name="AggregationType"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

<Parameters>

<Parameter Name="AggregationType">

<Value>3,4</Value>

</Parameter>

</Parameters>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.BooleanPicker

Description: A way to enter parameter value of Boolean type. Replaces standard SSRS Boolean picker

Properties:

Bindings

Default (report parameter has to be Boolean type)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.BooleanPicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="Enable3D"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DatePicker

Description: A way to select date value. Replaces standard SSRS date picker control.

Properties:

Bindings

Default (report parameter has to be DateTime type)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DatePicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />

<ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue"> <Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>

</ReportParameter>

<ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />

<ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />

<ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />

<ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>

</ReportParameter>

<ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />

<ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DateTimePicker

Description: A way to select date and time value.

Properties:

Bindings

Default (report parameter has to be DateTime type)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DateTimePicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="TimeZone" binding="TimeZone" />

<ReportParameter name="TimeZoneName" binding="TimeZoneName" />

<ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />

<ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>

</ReportParameter>

<ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />

<ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />

<ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />

<ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>

</ReportParameter>

<ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />

<ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.NumericUpDown

Description: A way to select an integer value.

Properties:

Bindings

Default (report parameter has to be Integer type)

Parameters

Minimum (minimum possible integer value)

Maximum (maximum possible integer value)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.DateTimePicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="TimeZone" binding="TimeZone" />

<ReportParameter name="TimeZoneName" binding="TimeZoneName" />

<ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />

<ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>

</ReportParameter>

<ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />

<ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />

<ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />

<ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>

</ReportParameter>

<ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />

<ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.TextBox

Description: A way to enter string parameter value. Replaces standard SSRS text box control.

Properties:

Bindings

Default

Parameters

Multiline (True/False)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.NumericUpDown" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="EventID"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

<Parameters>

<Parameter Name="EventID">

<Value>26319</Value>

</Parameter>

</Parameters>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.RelativeDateTimePicker

Description: A way to select relative date time range. This class is used on the report side to process values entered by this control.

Properties:

Bindings

TimeZone (time zone reference)

TimeZoneName (name of the time zone displayed in the report)

StartDate\_BaseType (start date base type)

StartDate\_BaseValue (start date base value)

StartDate\_OffsetType (type of start date offset)

StartDate\_OffsetValue (start date offset value)

EndDate\_BaseType (end date base type)

EndDate\_BaseValue (end date base value)

EndDate\_OffsetType (type of end date offset)

EndDate\_OffsetValue (end date offset value)

Parameters

MaxWidth (control maximum width in pixels)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.RelativeDateTimePicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="TimeZone" binding="TimeZone" />

<ReportParameter name="TimeZoneName" binding="TimeZoneName" />

<ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />

<ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>

</ReportParameter>

<ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />

<ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />

<ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />

<ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>

</ReportParameter>

<ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />

<ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.BusinessRelativeDateTimePicker

Description: A way to select relative date time range with business hours. This class is used on the report side to process values entered by this control.

Properties:

Bindings

TimeZone (time zone reference)

TimeZoneName (name of the time zone displayed in the report)

StartDate\_BaseType (start date base type)

StartDate\_BaseValue (start date base value)

StartDate\_OffsetType (type of start date offset)

StartDate\_OffsetValue (start date offset value)

EndDate\_BaseType (end date base type)

EndDate\_BaseValue (end date base value)

EndDate\_OffsetType (type of end date offset)

EndDate\_OffsetValue (end date offset value)

TimeType (type of time entered (business or regular))

TimeWeekMap (days of week selected for business time)

Parameters

MaxWidth (control maximum width in pixels)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.BusinessRelativeDateTimePicker" rowSpan="2" columnSpan="2">

<ReportParameters>

<ReportParameter name="TimeZone" binding="TimeZone" />

<ReportParameter name="TimeZoneName" binding="TimeZoneName" />

<ReportParameter name="StartDate\_BaseType" binding="StartDate\_BaseType" />

<ReportParameter name="StartDate\_BaseValue" binding="StartDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.StartDateTime</Prompt>

</ReportParameter>

<ReportParameter name="StartDate\_OffsetType" binding="StartDate\_OffsetType" />

<ReportParameter name="StartDate\_OffsetValue" binding="StartDate\_OffsetValue" />

<ReportParameter name="EndDate\_BaseType" binding="EndDate\_BaseType" />

<ReportParameter name="EndDate\_BaseValue" binding="EndDate\_BaseValue">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.EndDateTime</Prompt>

</ReportParameter>

<ReportParameter name="EndDate\_OffsetType" binding="EndDate\_OffsetType" />

<ReportParameter name="EndDate\_OffsetValue" binding="EndDate\_OffsetValue" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectPicker

Description: A way to select monitoring object for report. List of Monitoring Object IDs are passed to the report as a result value (default binding).

Properties:

Bindings

Default (report parameter has to be multi-value Integer type)

GroupList (list of management groups objects are allowed to be selected from)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="MonitoringObjectId" />

<ReportParameter name="ManagementGroupId" binding="GroupList" />

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectXmlPicker

Description: For object Selection input parameters as XML

Properties:

Bindings

Default (report parameter has to be multi-value Integer type)

GroupList (list of management groups objects are allowed to be selected from)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.MonitoringObjectXmlPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ObjectList</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceObjectPicker

Description: The Performance counter selection is based on the Object Selector. A performance counter selection has to support several objectives:

a) Scale is a parameter needed to allow comparing two counters with different value ranges.

b) The Dundas chart control allows changing of the chart type per series and also changing of the z-order (which series is sorted to the front, which to the back)

Properties:

Bindings

Default (report parameter has to be multi-value Integer type)

GroupList (list of management groups objects are allowed to be selected from)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceObjectPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ObjectList</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceChartObjectPicker

Description: For performance Chart Object Selection

Properties:

Bindings

Default (report parameter has to be multi-value Integer type)

GroupList (list of management groups objects are allowed to be selected from)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceChartObjectPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ObjectList"/>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ObjectList</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceRulePicker

Description: For Performance Rules Selection

Properties:

Bindings

Default (report parameter has to be multi-value Integer type)

GroupList (list of management groups objects are allowed to be selected from)

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceRulePicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ObjectList"/>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.PerformanceRule</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceRuleInstancePicker

Description: For Performance Rule Instances Selection

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.PerformanceRuleInstancePicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ObjectList"/>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.PerformanceRule</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.LinkedPerformanceObjectPicker

Description: For Performance Object Selection

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.LinkedPerformanceObjectPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ObjectList"/>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.PerformanceRule</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.LinkedPerformanceChartObjectPicker

Description: For performance Chart Object selection.

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.LinkedPerformanceChartObjectPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ObjectList"/>

<ReportParameter name="MonitoringObjectId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.PerformanceRule</Prompt>

</ReportParameter>

<ReportParameter name="ManagementGroupId" binding="GroupList"/>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.ReportColumnPicker

Description: For Report Column Selection

Sample:

<LinkedReport>

<ParameterBlock>

<Controls>

<Control type="Microsoft.SystemCenter.DataWarehouse.Report.ParameterControl.ReportColumnPicker" rowSpan="2" columnSpan="3">

<ReportParameters>

<ReportParameter name="ColumnId">

<Prompt>Microsoft.SystemCenter.DataWarehouse.Report.Library!Microsoft.SystemCenter.DataWarehouse.Report.ParameterPrompt.ColumnList</Prompt>

</ReportParameter>

</ReportParameters>

</Control>

</Controls>

</ParameterBlock>

</LinkedReport>

Appendix C - Data Types and Sample Queries

This appendix provides information about specific data types and sample queries for operational, performance, and discovery data that can be used when creating custom reports.

For State, Alert, and Performance data, you need to know which managed entity you want to query. To find this information, you get the ManagedEntityRowID from the vManagedEntity view. For groups, you can use the following query to return all managed entities within the group:

1. First, retrieve the ManagedEntityRowID for the group:  
select \* from vmanagedentity  
where fullname like 'Microsoft.SystemCenter.AllComputersGroup'

2. Note the ManagedEntityRowID.

3. Run the following query, with these changes:  
- Update the start(1st)and end(2nd) dates to return all entities that were a member of that group during these dates  
- Replace the Containment Value below (166) with the ManagedEntityRowID returned in Step 1.

EXEC [dbo].[Microsoft\_SystemCenter\_DataWarehouse\_Report\_Library\_ReportObjectLis tParse]'2008-01-01 10:50:23.150','2008-02-05 10:50:23.150', '<Data><Objects><Object Use="Containment">166</Object></Objects></Data>

4. You can join the resulting list of managed entities to the performance, state, and alert tables to retrieve relevant information.

Operational Data

State

State information is derived from different types of monitoring and is stored directly in the OperationsManagerDW database. State is summarized per object and per time, and it is made available in different time units: hourly, sub-hourly, and raw.

vStateHourlyFull

|  |  |
| --- | --- |
| Column | Datatype |
| DateTime | datetime |
| Date | datetime |
| Time | char(8) |
| ManagedEntityMonitorRowId | int |
| ManagedEntityRowID | int |
| IntervalDurationMilliseconds | int |
| InYellowStateMilliseconds | int |
| InRedStateMilliseconds | int |
| InPlannedMaintenanceMilliseconds | int |
| InUnplannedMaintenanceMilliseconds | int |
| InDisabledStateMilliseconds | int |
| HealthServiceUnavailableMilliseconds | int |
| InWhiteStateMilliseconds | int |
| InGreenStateMilliseconds | int |

vStateDailyFull

|  |  |
| --- | --- |
| Column | Datatype |
| DateTime | datetime |
| Date | datetime |
| Time | char(8) |
| ManagedEntityMonitorRowId | int |
| ManagedEntityRowID | int |
| IntervalDurationMilliseconds | int |
| InYellowStateMilliseconds | int |
| InRedStateMilliseconds | int |
| InPlannedMaintenanceMilliseconds | int |
| InUnplannedMaintenanceMilliseconds | int |
| InDisabledStateMilliseconds | int |
| HealthServiceUnavailableMilleseconds | int |
| InWhiteStateMilliseconds | int |
| InGreenStateMilliseconds | int |

vStateRaw

|  |  |
| --- | --- |
| Column | Datatype |
| EventOriginId | uniqueidentifier |
| ManagedEntityMonitorRowId | int |
| DateTime | datetime |
| OldHealthState | tinyint |
| NewHealthState | tinyint |

Query:

SELECT

vStateHourlyFull.DateTime,

vStateHourlyFull.InRedStateMilliseconds,

vStateHourlyFull.InYellowStateMilliseconds,

vStateHourlyFull.InGreenStateMilliseconds,

vStateHourlyFull.InWhiteStateMilliseconds,

vStateHourlyFull.InDisabledStateMilliseconds,

vStateHourlyFull.IntervalDurationMilliseconds,

vStateHourlyFull.InPlannedMaintenanceMilliseconds,

vStateHourlyFull.InUnplannedMaintenanceMilliseconds,

vStateHourlyFull.HealthServiceUnavailableMilliseconds,

vManagedEntity.ManagedEntityGuid,

vStateHourlyFull.ManagedEntityMonitorRowId,

vManagedEntityTypeImage.Image,

vManagedEntity.ManagedEntityDefaultName,

vManagedEntityType.ManagedEntityTypeGuid

FROM

vStateHourlyFull INNER JOIN

vManagedEntity ON vStateHourlyFull.ManagedEntityRowId =

vManagedEntity.ManagedEntityRowId INNER JOIN

vManagedEntityType ON vManagedEntity.ManagedEntityTypeRowId =

vManagedEntityType.ManagedEntityTypeRowId INNER JOIN

vMonitor ON vStateHourlyFull.MonitorRowId =

vMonitor.MonitorRowId LEFT OUTER JOIN

vManagedEntityTypeImage ON vManagedEntity.ManagedEntityTypeRowId

= vManagedEntityTypeImage.ManagedEntityTypeRowId AND

vManagedEntityTypeImage.ImageCategory = N'u16x16Icon'

WHERE

(vManagedEntity.ManagedEntityRowId in (/\*List of Managed Entities\*/)

AND (vMonitor.MonitorSystemName = @MonitorName)

AND (vStateHourlyFull.DateTime &gt;= @StartDate)

AND (vStateHourlyFull.DateTime &lt; @EndDate)

AND (vStateHourlyFull.Date

BETWEEN DATEADD(day, -1, @StartDate) AND DATEADD(day, 1, @EndDate))

Performance

Performance data is stored directly in the OperationsManagerDW. Performance data is available in daily and hourly aggregates, and with calculated minimum, maximum, and average values, and in the standard deviation.

Perf.vPerfDaily

|  |  |
| --- | --- |
| Column | Datatype |
| Datetime | datetime |
| PerformanceRuleInstanceRowID | int |
| ManagedEntityRowID | int |
| SampleCount | int |
| AverageValue | float |
| MinValue | float |
| MaxValue | float |
| StandardDeviation | float |

Perf.vPerfHourly

|  |  |
| --- | --- |
| Column | Datatype |
| Datetime | datetime |
| PerformanceRuleInstanceRowID | int |
| ManagedEntityRowID | int |
| SampleCount | int |
| AverageValue | float |
| MinValue | float |
| MaxValue | float |
| StandardDeviation | float |

Perf.vPerfRaw

|  |  |
| --- | --- |
| Column | Datatype |
| Datetime | datetime |
| PerformanceRuleInstanceRowID | int |
| ManagedEntityRowID | int |
| SampleCount | int |

Query:

SELECT

vManagedEntityTypeImage.Image,

vPerfHourly.DateTime,

vPerfHourly.SampleCount,

vPerfHourly.AverageValue,

vPerfHourly.StandardDeviation,

vPerfHourly.MaxValue,

vManagedEntity.FullName,

vManagedEntity.Path,

vManagedEntity.Name,

vManagedEntity.DisplayName,

vManagedEntity.ManagedEntityDefaultName,

vPerformanceRuleInstance.InstanceName,

vPerformanceRule.ObjectName,

vPerformanceRule.CounterName

FROM

Perf.vPerfHourly INNER JOIN

vManagedEntity ON Perf.vPerfHourly.ManagedEntityRowId =

vManagedEntity.ManagedEntityRowId INNER JOIN

vManagedEntityType ON vManagedEntity.ManagedEntityTypeRowId =

vManagedEntityType.ManagedEntityTypeRowId LEFT OUTER JOIN

vManagedEntityTypeImage ON vManagedEntityType.ManagedEntityTypeRowId =

vManagedEntityTypeImage.ManagedEntityTypeRowId INNER JOIN

vPerformanceRuleInstance ON

vPerformanceRuleInstance.PerformanceRuleInstanceRowId =

Perf.vPerfHourly.PerformanceRuleInstanceRowId INNER JOIN

vPerformanceRule ON vPerformanceRuleInstance.RuleRowId =

vPerformanceRule.RuleRowId

WHERE

(vManagedEntity.ManagedEntityRowId in (/\*List of Managed Entities\*/))

AND(vPerformanceRule.ObjectName IN ('MEMORY'))

AND (vPerformanceRule.CounterName IN ('Available MBytes'))

ORDER BY vPerfHourly.DateTime

Discovery

There are two types of data that are discovered: objects and views.

Objects

Operations Manager discovers the objects that have discovery rules defined for them. Discovery data is then stored in the OperationsManager and in the OperationsManagerDW databases. Besides up-to-date discovery data, the OperationsManagerDW database also contains historical discovery data even after the respective objects are no longer discovered. Using that data, you can then search the OperationsManagerDW database for objects for which the OperationsManager database no longer has active monitoring data.

Views

This view describes which management group the object belongs to. This view allows you to specify a range in which to search for an object. A single object can have multiple rows in vManagedEntity, as it can be discovered and removed multiple times in a distinct management group.

|  |  |
| --- | --- |
| Column | Datatype |
| ManagedEntityManagementGroupRowId | int identity |
| ManagedEntityRowId | int |
| FromDateTime | datetime |
| ToDateTime | datetime |
| DWCreatedDateTime | datetime |
| DWLastModifiedTime | datetime |

This view displays objects that were discovered by Operations Manager. The OperationsManagerDW contains data about objects that are no longer being actively managed by a management group. There can be multiple rows per managed entity because an entity can be discovered and removed within a management group multiple times.

|  |  |
| --- | --- |
| Column | Datatype |
| ManagedEntityRowId | int identity |
| ManagementGroupRowId | int |
| ManagedEntityGuid | uniqueidentifier |
| ManagedEntityTypeRowId | int |
| FullName | ntext |
| Path | ntext |
| Name | ntext |
| DisplayName | ntext |
| ManagedEntityDefaultName | ntext |
| DWCreatedDateTime | datetime |
| TopLevelHostManagedEntityRowId | int |

Object Properties

Object properties are discovered by discovery rules from different management packs. The OperationsManagerDW stores the discovery data and tracks when the discovery finds changes and what old and new values are.

Events

Events are stored directly in the OperationsManagerDW database with their respective details and parameters. No aggregation or summarization is done.

Event.vEvent

|  |  |
| --- | --- |
| Column | Datatype |
| EventOriginId | uniqueidentifier |
| DateTime | datetime |
| EventPublisherRowId | int |
| EventChannelRowId | smallint |
| EventCategoryRowId | smallint |
| EventLevelId | tinyint |
| LoggingComputerRowId | int |
| EventNumber | bigint |
| EventDisplayNumber | int |
| UserNameRowId | int |
| RawDescriptionHash | uniqueidentifier |
| ParameterHash | uniqueidentifier |
| EventDataHash | uniqueidentifier |

vEventCategory

|  |  |
| --- | --- |
| Column | Datatype |
| EventCategoryRowId | smallint identity |
| EventPublisherRowId | int |
| EventCategoryId | smallint |
| EventCategoryTitle | nvarchar |
| LastReceivedDateTime | smalldatetime |

vEventChannel

|  |  |
| --- | --- |
| Column | Datatype |
| EventChannelRowId | smallint identity |
| EventChannelTitle | nvarchar |
| LastReceivedDateTime | smalldatetime |

vEventDetail

|  |  |
| --- | --- |
| Column | Datatype |
| EventOriginId | uniqueidentifier |
| RawDescription | ntext |
| RenderedDescription | ntext |
| EventData | xml |

vEventParameter

|  |  |
| --- | --- |
| Column | Datatype |
| EventOriginId | uniqueidentifier |
| ParameterIndex | tinyint |
| ParameterValue | nvarchar |

vEventUserName

|  |  |
| --- | --- |
| Column | Datatype |
| EventUserNameRowId | int identity |
| UserName | nvarchar |
| LastReceivedDateTime | smalldatetime |

vEventLoggingComputer

|  |  |
| --- | --- |
| Column | Datatype |
| EventLoggingComputerRowId | smallint identity |
| ComputerName | nvarchar |
| LastReceivedDateTime | smalldatetime |

Query:

SELECT

      vEvent.DateTime,

      vEventPublisher.EventPublisherName as 'EventSource',

      vEventLoggingComputer.ComputerName as 'Computer',

      vEventLevel.EventLevelTitle as 'Type',

      vEvent.EventDisplayNumber as 'EventID',

      vEventChannel.EventChannelTitle,

      vEventUserName.UserName,

    vEventDetail.RenderedDescription as 'EventDescription'

FROM

      Event.vEvent LEFT OUTER JOIN

      vEventUserName ON vEvent.UserNameRowId =

vEventUserName.EventUserNameRowId LEFT OUTER JOIN

      vEventCategory ON vEvent.EventCategoryRowId =

vEventCategory.EventCategoryRowId LEFT OUTER JOIN

      vEventPublisher ON vEvent.EventPublisherRowId =

vEventPublisher.EventPublisherRowId LEFT OUTER JOIN

      vEventLoggingComputer ON vEvent.LoggingComputerRowId =

vEventLoggingComputer.EventLoggingComputerRowId LEFT OUTER JOIN

      vEventLevel ON vEvent.EventLevelId = vEventLevel.EventLevelId LEFT OUTER JOIN

      vEventChannel ON vEvent.EventChannelRowId =

vEventChannel.EventChannelRowId LEFT OUTER JOIN

      Event.vEventDetail ON vEvent.EventOriginId = vEventDetail.EventOriginId

WHERE vEventLevel.EventLevelTitle = 'Error'

ORDER BY vEvent.DateTime, vEventLoggingComputer.ComputerName

Alerts

Alerts are stored directly in the OperationsManagerDW database with their respective details and parameters. No aggregation or summarization is done.

Alert.vAlert

|  |  |
| --- | --- |
| Column | Datatype |
| AlertGuid | uniqueidentifier |
| AlertProblemGuid | uniqueidentifier |
| AlertName | nvarchar(256) |
| AlertDescription | nvarchar(256) |
| Severity | tinyint |
| Priority | tinyint |
| Category | nvarchar(256) |
| ManagedEntityRowID | int |
| WorkflowRowId | int |
| MonitorAlertInd | bit |
| RaisedDateTime | datetime |
| SiteName | nvarchar(256) |
| RepeatCount | int |
| AlertStringGuid | uniqueidentifier |
| ParameterHash | uniqueidentifier |
| DBCreatedDateTime | uniqueidentifier |
| DWCreatedDateTime | uniqueidentifier |
| DWLastModifiedTime | uniqueidentifier |

Alert.vAlertDetail

|  |  |
| --- | --- |
| Column | Datatype |
| AlertGuid | uniqueidentifier |
| Owner | nvarchar(256) |
| TicketID | nvarchar(256) |
| CustomField1 | nvarchar(256) |
| CustomField2 | nvarchar(256) |
| CustomField3 | nvarchar(256) |
| CustomField4 | nvarchar(256) |
| CustomField5 | nvarchar(256) |
| CustomField6 | nvarchar(256) |
| CustomField7 | nvarchar(256) |
| CustomField8 | nvarchar(256) |
| CustomField9 | nvarchar(256) |
| CustomField10 | nvarchar(256) |
| DBLastModifiedDateTime | datetime |
| DBLastModifiedByUserId | nvarchar(256) |
| DWCreatedDateTime | datetime |

Alert.vAlertParameter

|  |  |
| --- | --- |
| Column | Datatype |
| AlertGuid | uniqueidentifier |
| ParameterIndex | tinyint |
| ParameterValue | nvarchar(255) |

Query:

SELECT

Alert.vAlert.AlertName,

Alert.vAlert.AlertDescription,

Alert.vAlert.Severity,

Alert.vAlert.Priority,

Alert.vAlert.Category,

Alert.vAlert.RaisedDateTime,

Alert.vAlert.RepeatCount,

vManagedEntity.FullName,

vManagedEntity.Path,

vManagedEntity.Name,

vManagedEntity.DisplayName,

vManagedEntity.ManagedEntityDefaultName

FROM Alert.vAlertDetail INNER JOIN

Alert.vAlert ON Alert.vAlertDetail.AlertGuid = Alert.vAlert.AlertGuid INNER JOIN

vManagedEntity ON Alert.vAlert.ManagedEntityRowId =

vManagedEntity.ManagedEntityRowId

WHERE vManagedEntity.ManagedEntityRowId in (/\*List of Managed Entities\*/)

AND Alert.vAlert.Severity = 1

Note: The queries above will return values for images which will allow you to visualize an object in a report.

Visualizing the objects shown using an image

Where is it used?

When an object is used in a report the image is queried from the vImage view.

How is it defined in the report?

<Image Name="ObjectImage">

<Sizing>Clip</Sizing>

<MIMEType>image/png</MIMEType>

<ZIndex>7</ZIndex>

<Source>Database</Source>

<Style>

<BorderStyle>

<Default>Solid</Default>

<Right>None</Right>

</BorderStyle>

<PaddingLeft>4pt</PaddingLeft>

<BorderColor>

<Default>DarkGray</Default>

</BorderColor>

<PaddingTop>2pt</PaddingTop>

</Style>

<Value>=Fields!Image.Value</Value>

</Image>

Where does the image come from?

The image is taken from a management pack