# logo-sql08.gif

Policy-Based Management

SQL Server Technical Article

**Writer:** Sharon Dooley, SQL Server consultant and author

**Technical Reviewers:** Hongfei Guo and Dan Jones

**Published:** June 2009

**Applies to:** SQL Server 2008

**Summary:** This paper explores Policy-Based Management, a new management feature introduced in SQL Server 2008. Policy-Based Management allows DBAs to define a set of policies that can control many different aspects of SQL Server. Policies can be applied to a single server or to groups of servers. For example, a DBA could define a policy that specifies how a particular configuration option should be set on all the servers in the enterprise. What used to be time-consuming tasks and referencing paper manuals can now be virtually automated.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

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.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, PowerShell, SQL Server, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

[Introduction 5](#_Toc231789780)

[What is Policy-Based Management? 5](#_Toc231789781)

[Why Use Policy-Based Management? 5](#_Toc231789782)

[Terminology 5](#_Toc231789783)

[Policies 5](#_Toc231789784)

[Conditions 5](#_Toc231789785)

[Facets 5](#_Toc231789786)

[Evaluation Modes 6](#_Toc231789787)

[Targets 6](#_Toc231789788)

[Category 6](#_Toc231789789)

[Getting Started with Policy-Based Management 7](#_Toc231789790)

[Defining Policies 7](#_Toc231789791)

[Creating a Category 8](#_Toc231789792)

[Creating a Simple Condition 8](#_Toc231789793)

[Creating a Policy 9](#_Toc231789794)

[Testing the Policy 11](#_Toc231789795)

[Getting Notifications When Policies Are Evaluated 11](#_Toc231789796)

[Selecting the Right Facet 12](#_Toc231789797)

[Policy-Based Management System Views 14](#_Toc231789798)

[Importing and Exporting Policies 17](#_Toc231789799)

[Evaluating Policies on Multiple SQL Servers 17](#_Toc231789800)

[Evaluating Policies on a Registered Server 17](#_Toc231789801)

[Evaluating Policies with Central Management Servers 18](#_Toc231789802)

[Best Practices Policies 20](#_Toc231789803)

[Importing the Best Practices Policies 20](#_Toc231789804)

[Best Practices Policies Categories 20](#_Toc231789805)

[Constructing Advanced Conditions 21](#_Toc231789806)

[ExecuteSql 24](#_Toc231789807)

[ExecuteWql 25](#_Toc231789808)

[Surface Area Configuration 26](#_Toc231789809)

[Policy-Based Management and Prior Versions of SQL Server 26](#_Toc231789810)

[Enterprise Policy Management Framework (EPM) 26](#_Toc231789811)

[Summary and Conclusions 26](#_Toc231789812)

[Appendix: Policy-Based Management Facets 28](#_Toc231789813)

[About the Author 55](#_Toc231789814)

# Introduction

## What is Policy-Based Management?

Policy-Based Management is a new feature in SQL Server 2008 that allows administrators to define rules for one or more SQL Servers and enforce them. The goal of this feature is to make it easier for an administrator to manage one or more servers by preventing servers from being out of compliance with his or her policies. Policies can be used in a variety of ways. For example, you can use policies to enforce naming conventions in a database. You may have several servers that you want to have the same settings of various configuration options. Policy-Based Management will allow you to discover when one of those servers goes out of compliance. Even the use of SQL Server features can be managed with Policy-Based Management – you can use Policy-Based Management to ensure that SQLMail is not enabled on any server in your enterprise. Policies can be applied to a single server or many servers.

## Why Use Policy-Based Management?

The only tool that database administrators have had in the past to control the setup of servers and databases is a paper-based policy manual. Discovering policy violations and who was responsible for them was a manual task. It was often difficult to discover why a policy had been violated, and even more difficult to determine who violated it. Making sure that a large number of servers were all set up in the same way was a major headache.

The introduction of Policy-Based Management in SQL Server 2008 solves this problem and can be a significant time saver. It is now possible to define how servers are to be configured and have SQL Server reason over these policies to enforce the rules.

## Terminology

Before we cover the ins and outs of using Policy-Based Management, it’s important to understand a few core concepts related to PBM. Figure 1 below illustrates the relationship of these components within PBM.

### Policies

A policy is a rule based on a single condition and applied to one or more targets. A policy has an automation mode that describes what SQL Server should do when a policy is violated.

### Conditions

A condition contains one or more Boolean expressions that can be combined with AND, OR, and NOT. The conditions can be quite complex.

### Facets

A facet is a collection of properties for an object such as a table, a stored procedure, or an audit. A facet’s properties are used to test various conditions. For example, the Table facet contains name, file group, owner, createdate, and hasclusteredindex properties along with many others. Facets are pre-defined and the set of available facets cannot be extended in SQL Server 2008.

### Evaluation Modes

There are four choices for policy evaluation:

On Change: Prevent

On Change: Log only

On Schedule

On Demand

The on change: prevent mode uses DDL triggers to prevent undesirable changes from happening. (Note that the server must allow nested triggers for this mode to work.) On change: log allows the change to be made but records it in the event log. The On change evaluation modes are only available for those changes that can be trapped with a DDL trigger. On schedule evaluates policy compliance on a given schedule. Compliance or non-compliance is reported in the policy history. On demand requires that the administrator manually request policy evaluation. The on change: prevent, on schedule, and on demand modes will record errors in the event log. An administrator can create alerts that respond to the error and send notifications about the policy violations.

Some facets do not support all the evaluation modes. At the end of this paper, there is a chart that shows all the facets, their supported evaluation modes, their targets, and their available properties.

### Targets

A target is a SQL Server instance, one or more databases, one or more tables, or one or more indices. The targets form a hierarchy. If a policy is created at the server level, it applies to all the appropriate lower levels. A target is also called an “object set”.

### Category

You can use categories to manage policies. Categories can be used simply to group related policies, but there are more powerful uses for them as well. If you specify that a particular category of policies is mandated for all databases, these policies will be applied to all databases whether or not the database owner wants them. If a category is not mandated to all databases, a database owner can subscribe to one or more policy categories. There is always a default category; policies not otherwise categorized are placed in this category and are applied to all databases.

**Figure 1: The various components of Policy-Based Management**

Category (optional)

Policy

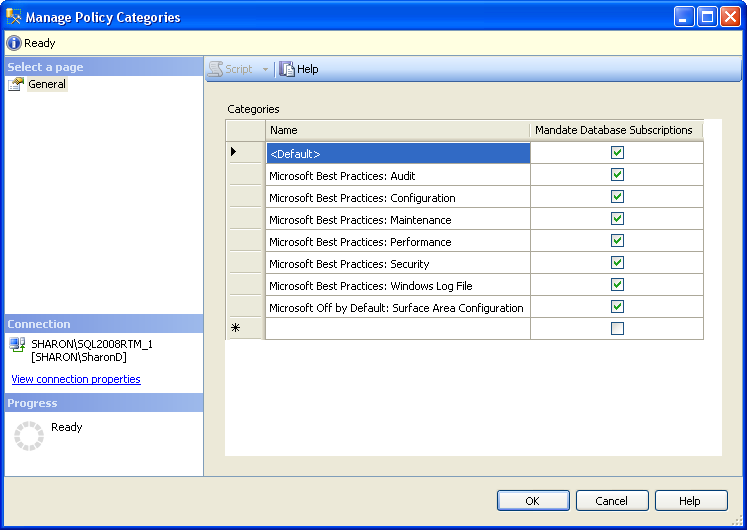
Condition

Expression

Property

Facet

1..\*

1

1..\*

1..\*

1..\*

Target (ObjectSet)

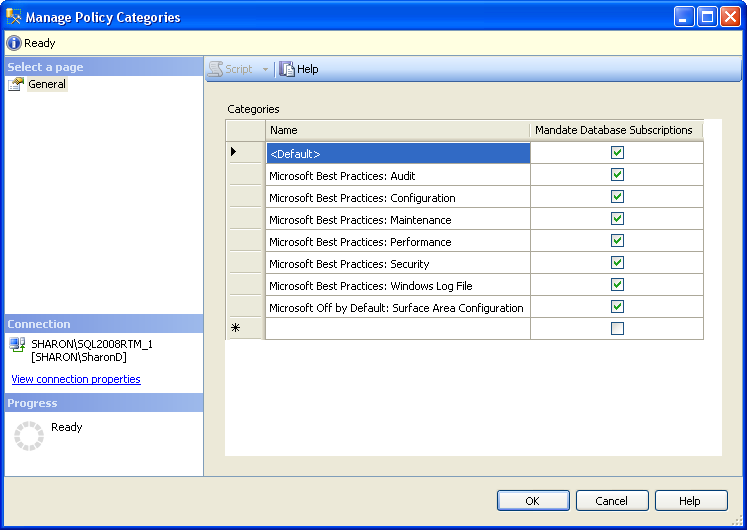
Subscribers

0..\*

1..\*

Evaluation mode

1

1

# Getting Started with Policy-Based Management

## Defining Policies

The easiest way to define policies is to use the graphical tools in SQL Server Management Studio. While it is possible to create policies with Transact-SQL, Microsoft does not recommend this.

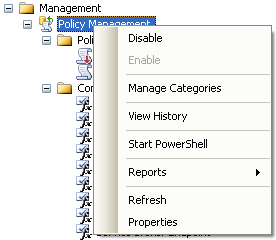
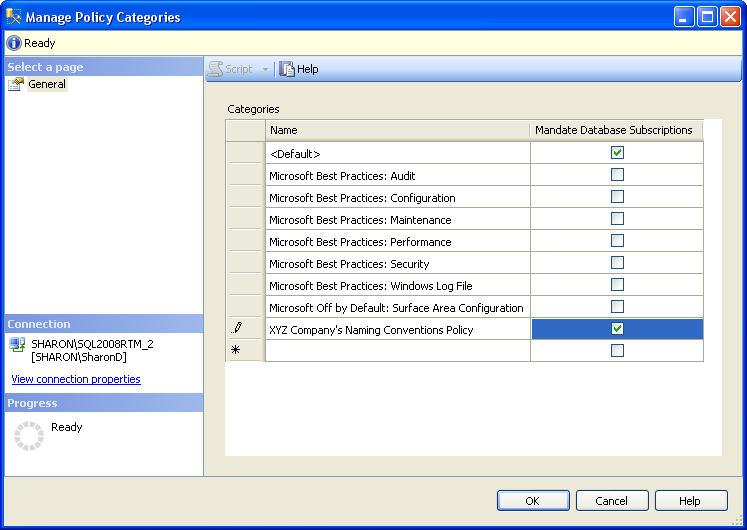
If you need to manipulate policies programmatically, you should use a .NET language and the policy management objects in the SQL Server Management Objects API. The top level object in this API is the PolicyStore class. It has the following child objects:

* Policy,
* Condition,
* PolicyGroupSubscription,
* PolicyCategory, and
* Facets

Operations such as moving policies from a test environment to a production environment are best done by using the export and import feature.

The following sections show you how to use the GUI to create a simple policy.

### Creating a Category

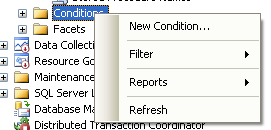
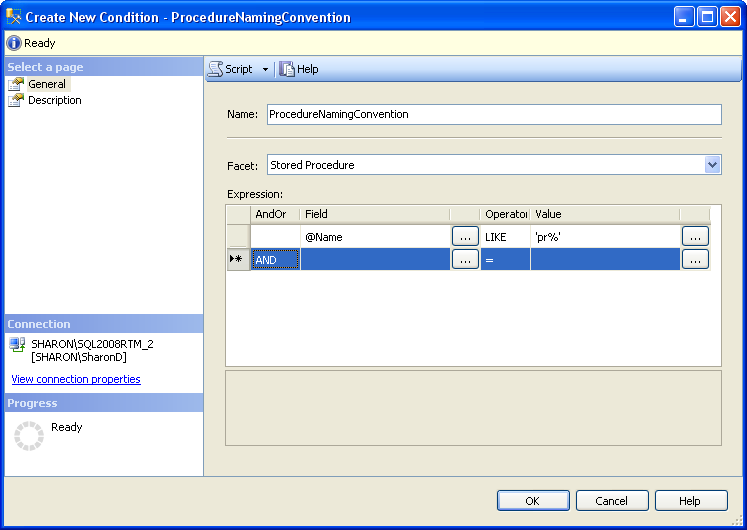
 

Creating a category is as simple as it looks. Enter the category name and select whether you want the policy to apply to all databases.

### Creating a Simple Condition

A policy must have a condition. You can create the condition first from the Conditions context menu as shown here. You can also simply start with the policy, and create the policy from there.

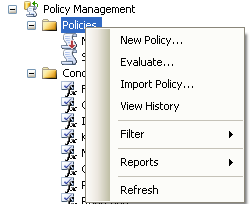
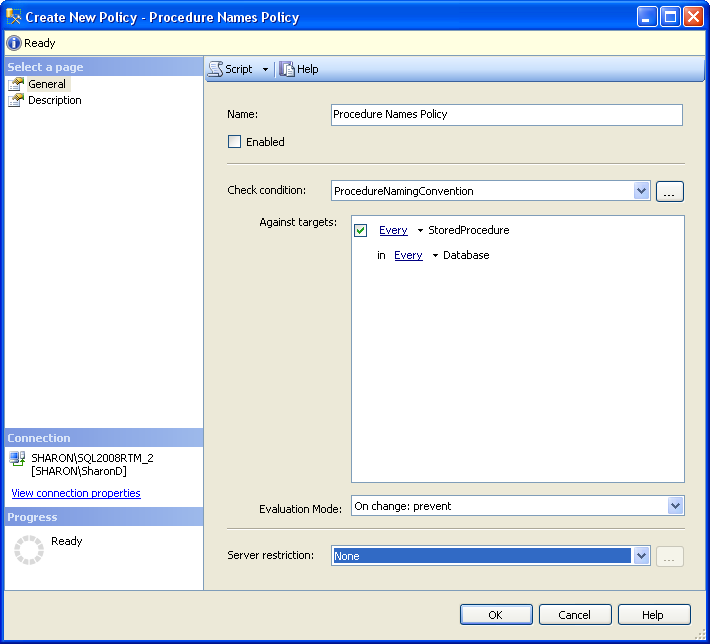
This example will create a condition that prohibits creation of stored procedures with names that don't begin with "pr".

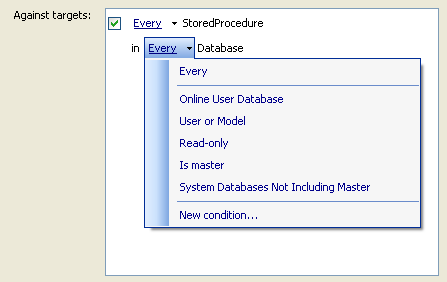
You provide a name for the condition. Then you select the appropriate facet from the drop-down. In this case, since we want to require that procedure names start with pr, we will use the stored procedure facet. (See the section titled “Selecting the right facet” for other facets that you might use. Once you have selected a facet, the Fields drop down will show the available fields for that facet and the list of operators appropriate for that field will be populated. The Description page allows you to document the condition.

### Creating a Policy

This example will create a policy that can be used to enforce the procedure naming convention condition that we created above.

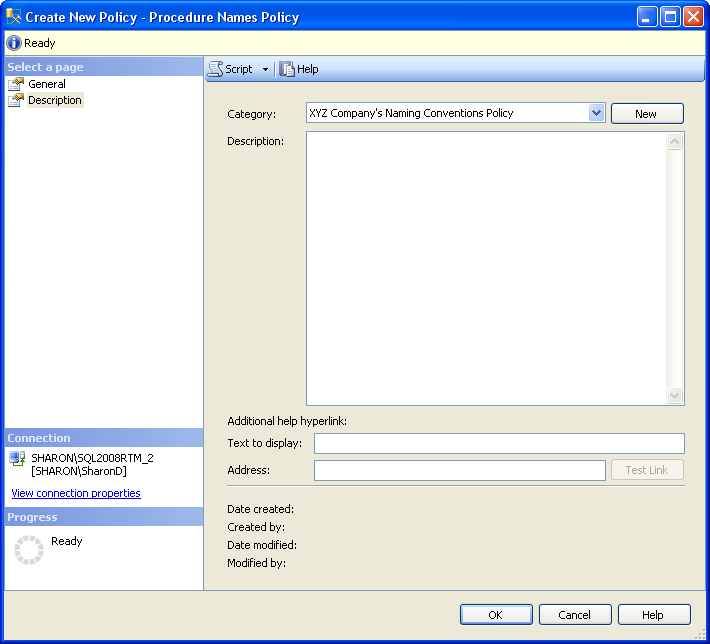
 

You provide a name for the policy and select the condition from the drop-down. Then you specify the targets. This example uses every stored procedure in every database, but you have a variety of choices, including the ability to define a condition that specifies the target.



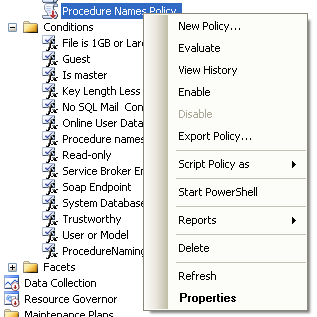
Finally you specify the Evaluation mode. In this example, we'll use On change: prevent.

If you want to specify a category for the policy, you use the Description page and select the desired category from the drop-down.

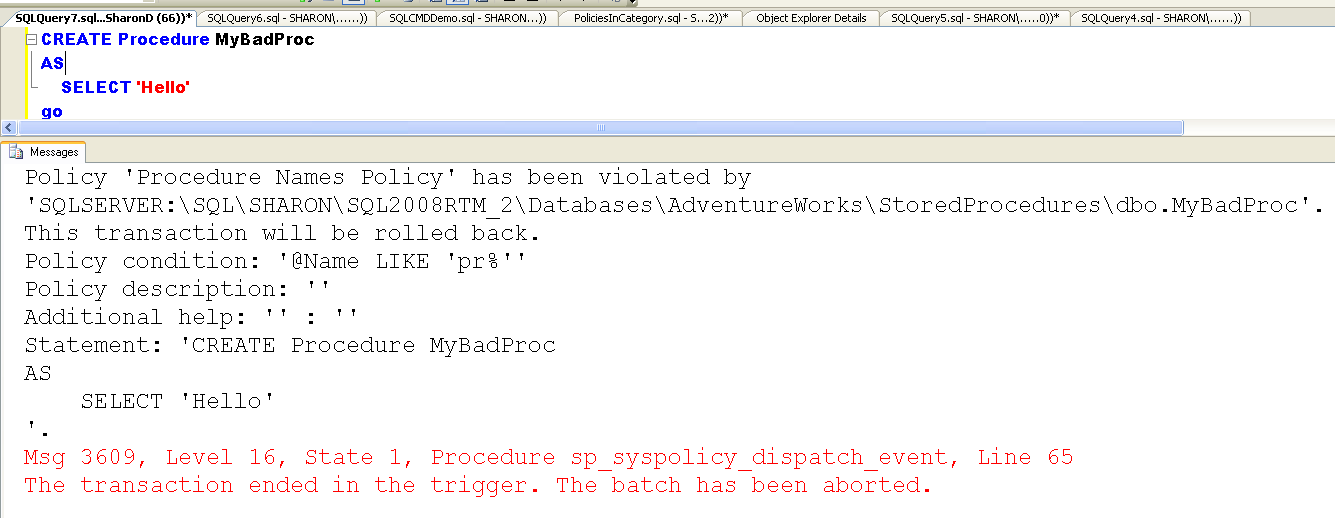


### Testing the Policy

In order to see how the On change: Prevent evaluation mode we’ve selected, we need to enable it.



Once that’s done, seeing the effect of an On change: Prevent policy is simply a matter of creating an object that violates the policy and making sure that it is rejected and creating one that conforms and making sure that it is accepted.



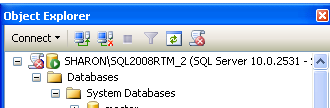
### Getting Notifications When Policies Are Evaluated

SQL Server Agent’s alerting system is easily tied to Policy-Based Management. The error message numbers are different for each of the evaluation modes and are shown below.

|  |  |
| --- | --- |
| Message Number | Evaluation Mode |
| 30450 | On change: Prevent when the policy is enabled for automatic evaluation |
| 30451 | On change: Prevent when the policy is set to on demand evaluation |
| 30452 | On schedule |
| 30453 | On change: Log |

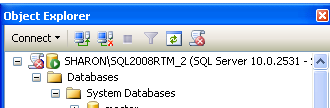
Notice that the error numbers associated with Policy-Based Management are not the ones you’ll see in the messages window. The violation demonstrated above ended with error 3609. However, this error message could be caused by other DDL trigger operations. To capture this policy violation, you need to tie the alert to Message Number 30450.

You can also see if there are any current policy violations with Management Studio. Once policies have been evaluated, in Object Explorer, you will see an icon next to the server name if there are any out-of-compliance situations. This icon will only show the results for enabled policies.



This tells you that there is a policy violation somewhere in the server.

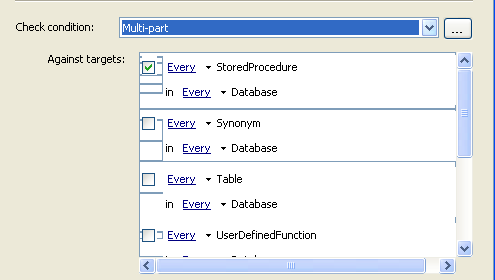
If you have evaluated policies on a server, and you know there were out-out of compliance issues but don’t see the icon, perhaps the toggle button in the tool bar has been inadvertently clicked. To restore the display of the individual server icons, click the button circled below.



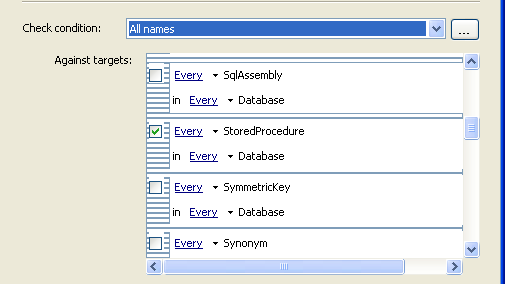
# Selecting the Right Facet

If you look at the Facet chart provided at the end of this paper, you’ll see there are hundreds of facets and that many of them seem to overlap. You’ll need to select the appropriate facet for each condition that you create.

In the example above, we used the Name property of the Stored Procedure facet. There are two other facets that we could have used to accomplish the same result: Multipart Name and Name. The Multipart Name facet allows On Change evaluation; the Name facet does not. Where you would see a difference if you used either of those facets is in the set of available targets. The Multipart Name facet includes common database objects as targets.



The Name facet gives you access to a large group of possible targets.



Suppose that, instead of specifying a specific naming convention for stored procedures, the XYZ company wanted all tables, stored procedures, synonyms, user defined functions, user defined types, views, and XmlSchemaCollections to start with the string “xyz” and that they wanted to use the On Change: Prevent evaluation mode. The Multipart Name facet would allow them to do this. If they wanted, instead, to have a policy that all namable items need to start with “xyz”, the name facet would be the easiest to use. However, they would have to give up On Change: Prevent evaluation.

In some cases, the combination of evaluation mode and targets may determine what facet you choose. In other cases, the evaluation mode will be the only reason for making a decision. For example, both the Database facet and the Database Options facet have all of the database options as properties. Only the Database Options facet provides for On Change evaluation. If you wanted to set a policy that specified that the auto close option must always be false, you could use either facet. If you wanted to prevent anyone from setting the option to true, you’d have to use the Database Options facet.

# Policy-Based Management System Views

As it does for other aspects of SQL Server, Microsoft provides seven system views that allow you to obtain information about policies. I find these views extremely useful in many places, and Policy-Based Management is no exception. It's difficult to form queries without knowing how the views relate to each other. Figure 2 shows the relationships between the views.

syspolicy\_policy\_categories

syspolicy\_policies

syspolicy\_conditions

syspolicy\_policy\_execution\_history

syspolicy\_policy\_execution\_history\_details

syspolicy\_system\_health\_state

syspolicy\_policy\_category\_subscriptions

One-to-many relationship

syspolicy\_configuration

syspolicy\_object\_sets

syspolicy\_target\_sets

**Figure 2: Policy-Based Management System Views**

Although you can define categories with the graphical tools and place policies in those categories, you can’t see a list of all the policies in a given category. Here is a query that will produce the list:

SELECT CASE

WHEN c.name IS NULL THEN 'Default'

ELSE c.name

END as CategoryName

, p.Name as PolicyName

, CASE mandate\_database\_subscriptions

WHEN 0 THEN 'No'

ELSE 'Yes'

END AS 'Mandated?'

FROM syspolicy\_policy\_categories as c

FULL OUTER JOIN syspolicy\_policies as p

ON c.policy\_category\_id = p.policy\_category\_id

ORDER BY CategoryName

We can extend this query so that we see the results of the most recent evaluation of every policy in every category:

SELECT CASE

WHEN c.name IS NULL THEN 'Default'

ELSE c.name

END as CategoryName

, p.Name as PolicyName

, CASE mandate\_database\_subscriptions

WHEN 0 THEN 'No'

ELSE 'Yes'

END AS 'Mandated?'

, target\_query\_expression

, CASE

WHEN result IS NULL THEN 'N/A'

WHEN result = 1 THEN 'OK'

ELSE 'Bad'

END AS 'Last evaluation result'

FROM syspolicy\_policy\_categories as c

FULL OUTER JOIN syspolicy\_policies as p

ON c.policy\_category\_id = p.policy\_category\_id

LEFT JOIN syspolicy\_system\_health\_state AS h

on p.policy\_id = h.policy\_id

ORDER BY CategoryName

The following script will show you the target for the policy that was created in the demonstration earlier in this paper:

SELECT p.name, t.type\_skeleton AS Target

FROM dbo.syspolicy\_policies AS p

INNER JOIN dbo.syspolicy\_object\_sets as o

ON p.object\_set\_id = o.object\_set\_id

INNER JOIN dbo.syspolicy\_target\_sets as t

ON o.object\_set\_id = t.object\_set\_id

WHERE p.policy\_id = 54

And here’s a query that will show you all the databases that have subscribed to the policy categories:

SELECT c.Name as CategoryName

, target\_object AS DatabaseName

FROM syspolicy\_policy\_categories as c

LEFT OUTER JOIN   
syspolicy\_policy\_category\_subscriptions as s

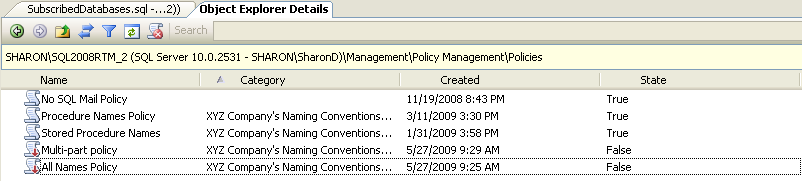
ON c.policy\_category\_id = s.policy\_category\_id

ORDER BY CategoryName, DatabaseName

# Importing and Exporting Policies

If you need to move policy definitions from one server to another, it’s a simple process to export them, copy them to their destination, and import them. Right-click on a Policy in Management Studio to export it to the file system. To import one or more policies, right-click on the Policies folder.

If you work in the Object Explorer window, it appears that you can only export one policy at a time. But if you use the Object Explorer Details window, you’ll be able to export many policies at once. For example, if you want to export all the policies in a specific category, click the Category label at the top of the display to sort the policies into category order.



Then, select all the policies in the category (the standard click on the first one and shift-click on the last one works), right-click and select Export.

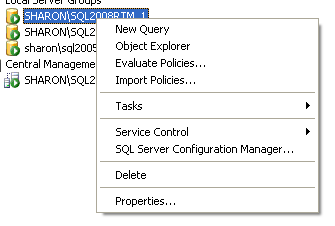
# Evaluating Policies on Multiple SQL Servers

DBAs are often faced with trying to make sure configuration and other settings are the same on multiple servers in the enterprise. If you define your configuration policies with Policy-Based Management this task will be much easier.

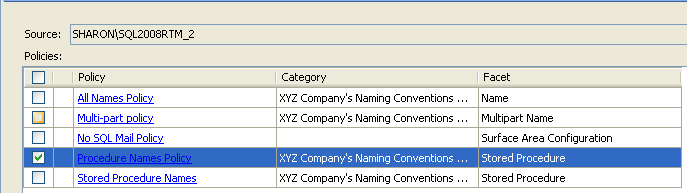
There are two ways to evaluate policies defined on one server against another server. You can do it through the Registered Servers window in Management Studio or you can designate Central Management Servers.

## Evaluating Policies on a Registered Server

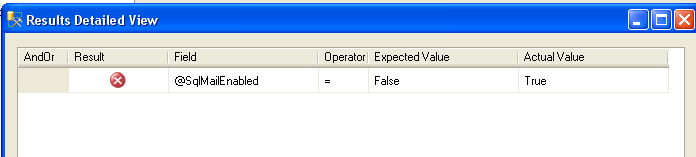
It’s easy to evaluate policies against any server that’s in your registered servers window. Simply right-click the server and choose Evaluate Policies.



Specify the whether you want to use policies that are stored in the file system or policies that are defined on a specific SQL Server, then select the policy(ies) that you want to evaluate.



When you click the Evaluate button, you will see the overall results of the evaluation and that result will give you an opportunity to see the details.



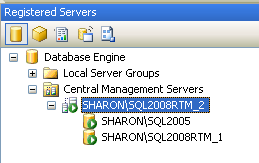
## Evaluating Policies with Central Management Servers

SQL Server 2008 introduces something called a “Central Management Server” which serves two different functions:

1. The same query can be run against all of the servers
2. Policies can be evaluated against all of the servers

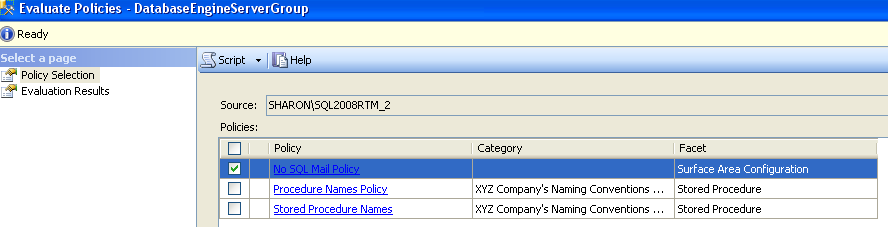
If you need to evaluate a set of policies against a group of servers, you will find it easy to accomplish with a Central Management Server. It’s important to understand that the Central Management Server won’t participate in the evaluation – the evaluation will be done only on the servers that are being managed. If you are planning to use Central Management Servers, consider using stand-alone servers, not one of your production servers. Define the policies on the Central Management Server and use it to control the evaluation of the policies on the other serves.

Central Management Servers are defined in the Registered Servers window in Management Studio. You designate one server as the Management Server and register the servers you want to manage with it. The Management Server itself is not a member of the group. In the illustration below, Sharon\SQL2008RTM\_2 is the management server for Sharon\SQL2005 and Sharon\SQL2008RTM\_1.

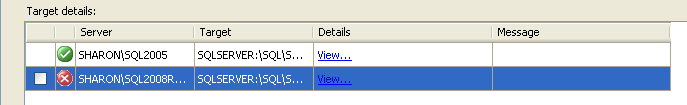


The process of registering the servers with a Central Management Server is almost identical to the process of registering servers in Management Studio.

To evaluate policies on the servers managed by the Central Management Server, right-click, select “Evaluate policies” and specify the server where the policies are defined in the dialogue.



Then select the policy(ies) you want to test. After evaluation, you will see the results.



You will probably want your central management server to be one that is not running your production workload.

# Best Practices Policies

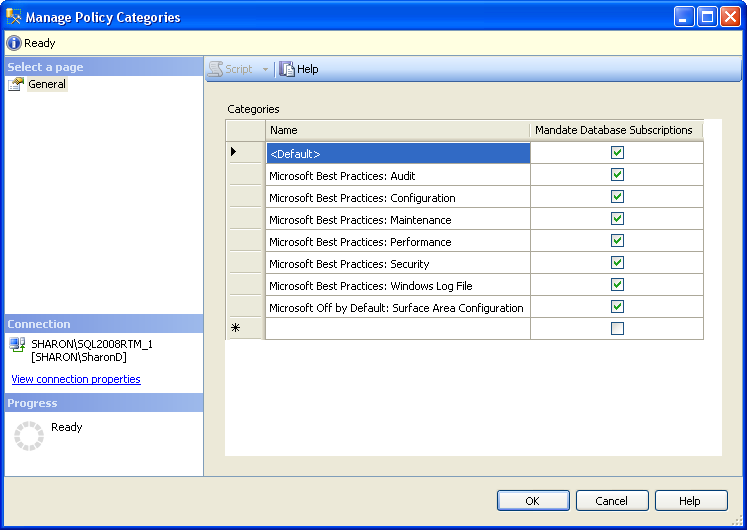
I have always been a fan of Microsoft’s Best Practices Analyzer. It helped me to make sure that my databases had not deviated from these best practices. In SQL Server 2008, the Best Practices Analyzer has been replaced by best practices policies. You might find this less easy to use than the old stand-alone application. But having it in the environment where you do so much other DBA work is worth the tradeoff.

## Importing the Best Practices Policies

The best practices policies are not installed in SQL Server by default. However, they do ship with SQL Server 2008, and the Database Engine policies are in C:\Program Files\Microsoft SQL Server\100\Tools\Policies\DatabaseEngine\1033. There are also best practices policies for Analysis Services and Reporting Services.

## Best Practices Policies Categories

The best practices policies are organized into categories as shown below.



By default, they are mandated in all databases, but you can change this if you want individual database owners to be able to subscribe to them for their databases. None of the best practices policies are enabled when you import them. All of them are set for On Demand evaluation.

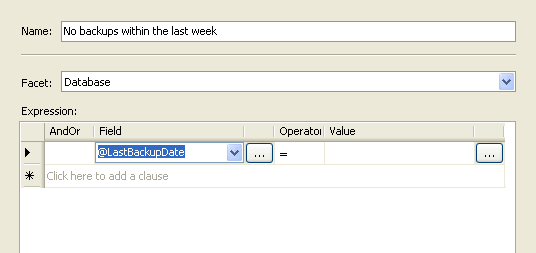
# Constructing Advanced Conditions

The facets and properties provide a rich set of SQL Server elements for use in Policy-Based Management conditions. But the choices in the value drop-down, such as True and False, may not always meet your requirements. Similarly, the pre-defined list of targets may not meet your needs. Fortunately, Policy-Based Management allows you to define virtually any condition or target you can think of. However, you must do this with an expression language that is not the familiar Transact-SQL used by most DBAs. You will find that even where the functions have familiar names, such as DateAdd, their syntax is very different. First, assume that the function names and all of their arguments are case-sensitive. There are also other differences. Consider the DateAdd function:

Syntax: DateTime DateAdd(String datepart, Numeric number, DateTime date)

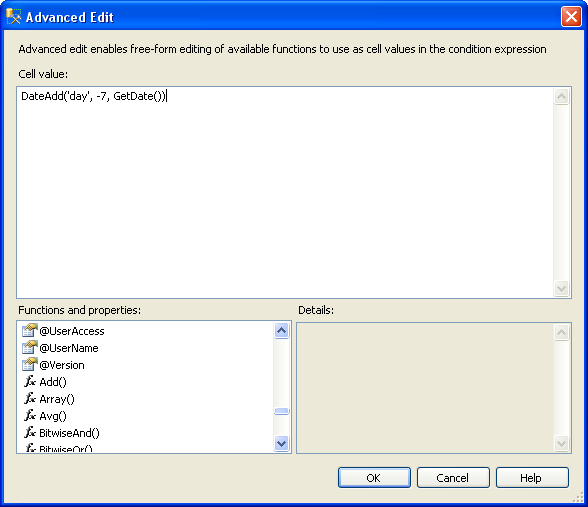
Example: SELECT DateAdd('day', 21, DateTime('2007-08-06 14:21:50')) returns '2007-08-27 14:21:50'

Assume you need to create a policy that will tell you which databases have not been backed up within the last week. Also, you only want to apply this policy to databases that are 50 MB or larger. You’ll need to use the Database facet and the LastBackupDate property.



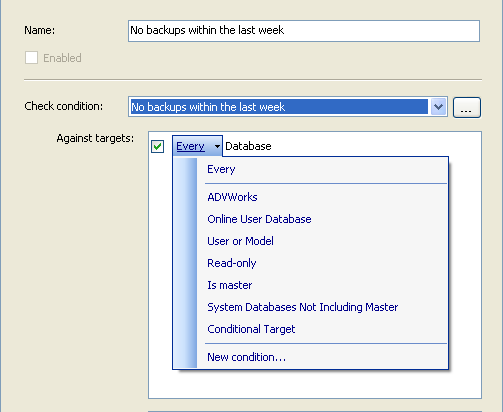
Then you need to click the browse button to the right of the Value column. This will bring you to the advanced edit dialogue. Here you can enter whatever expression you need. In this example, the expression is:

DateAdd('day', -7, GetDate())

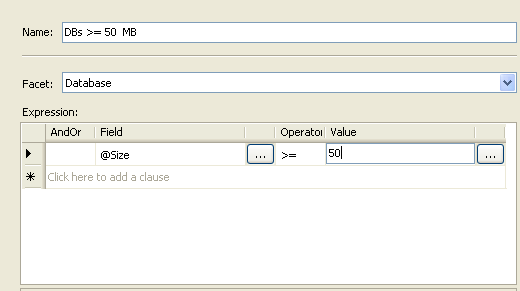


For help on any function that you want to use, highlight it. The syntax, an explanation and an example, will be shown in the details box.

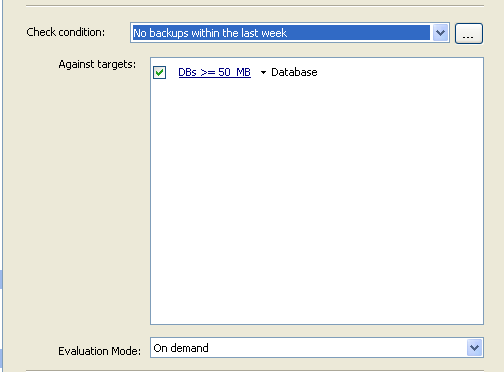
To implement the conditional target for the Policy, click the drop-down arrow next to “Every”.



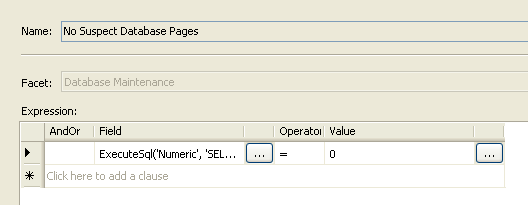
Select “New Condition”.



Your policy will show your conditional target.



You can also use an expression for the Field column. Here is one of the best practices policies.



The ExecuteSQL function takes us back into the familiar Transact-SQL world:

ExecuteSql('Numeric', 'SELECT COUNT(\*) AS [Total\_Suspect\_Pages] FROM msdb.dbo.suspect\_pages WHERE event\_type IN (1,2,3) AND database\_id = DB\_ID(DB\_NAME()) ')

### ExecuteSql

If you can write a SQL query that defines the desired state of some aspect of your SQL Server, you can create a condition that allows you to evaluate whether your servers are in compliance with that state by using the ExecuteSql function.

The ExecuteSql function has two arguments: a character string that defines the type of data being returned and a character string that contains the SQL query.

The SQL statement can contain any valid SQL statement including data modification statements, create statements and grant statements.

The ExecuteSql function will return only the first result returned by the query.

When a condition contains the ExecuteSql function, it will be evaluated against all of the targets for the policy. For example, if the target is the server, it will run once. If the target is all databases, it will run once for each database, and if the target is an object class such as a stored procedure, it will run once for each stored procedure.

ExecuteSQL queries can contain two “placeholder” variables: @SchemaName and @ObjectName. You can use these in the queries in any place where you would normally provide an object name. If you write SELECT … FROM @SchemaName.@ObjectName, the actual names will be substituted in the query.

There are security issues that you’ll need to consider when you’re using ExecuteSql. For information about this topic and an example of a complex ExecuteSql query, visit <http://blogs.msdn.com/sqlpbm/> and locate the post titled ExecuteSql().

### ExecuteWql

SQL Server has some built-in support for the Windows Management Infrastructure (WMI). WMI is designed to allow Windows administrators to manage their Windows servers programmatically. A subset of SQL can be used to query WMI. SQL Server Administrators primarily use this capability to obtain information about the operating system. Sometimes, it may be useful to use WMI in defining conditions and the ExecuteWQL function allows you to do exactly that. In order to take advantage of this capability, you’ll need to understand the WMI objects and namespace.

The ExecuteWql function has three arguments: a character string that defines the data type that’s being returned by the query, a character string that identifies the WMI Namespace you are using, and a character string that contains the actual WQL query.

With ExecuteWql, you could define a condition that requires that a particular drive has at least 10000 mb of free space. Evaluation of the policy that includes this condition would show you the drives that had less space available. The expression you would use for the field is:  
 ExecuteWql('Numeric', 'Root\CIMV2', 'select FreeSpace from Win32\_LogicalDisk')

When you define the field with an expression, the facet selected becomes irrelevant. However, you should select one that provides the evaluation mode you want to use with the policy.

# Surface Area Configuration

SQL Server 2005 had a tool named “SQL Server Surface Area Configuration”. This tool was used to manage services and to control which initially disabled features were activated. The services part of this application primarily determined what network protocols SQL Server was listening. Both of the functions this tool performed could be done without it. Management of network protocols could be done with SQL Server Configuration Manager and enabling and disabling features could be done with sp\_configure.

SQL Server 2008 does not have a Surface Area Configuration tool. But it does have a facet named “Surface Area Configuration”. You can use this facet and develop policies that will let you know if there are any servers that have features enabled that aren’t supposed to be enabled in your organization. The Surface Area Configuration facet only supports On Demand evaluation.

# Policy-Based Management and Prior Versions of SQL Server

Policies must be created with SQL Server 2008 tools and must reside on SQL Server 2008 or in the file system. However, it is possible to use the Registered Servers window (with or without Central Management Servers) to evaluate policies against a SQL Server 2005 or SQL Server 2000 instance. Only On Demand evaluation is available for earlier versions of SQL Server.

# Enterprise Policy Management Framework (EPM)

The Enterprise Policy Management Framework (EPM) is a solution to extend SQL Server 2008 Policy-Based Management to all versions of SQL Server in an enterprise, including SQL Server 2000 and SQL Server 2005. The EPM Framework will report the state of specified SQL Server instances against policies that define the defined intent, desired configuration, and deployment standards.

When the Enterprise Policy Management Framework (EPM) is implemented, policies will be evaluated against specified instances of SQL Server through Windows PowerShell. This solution will require at least one instance of SQL Server 2008. The PowerShell script will run from this instance through a SQL Server Agent job or manually through the PowerShell interface. The PowerShell script will capture the policy evaluation output and insert the output to a SQL Server table. SQL Server 2008 Reporting Services reports will deliver information from the centralized table.

Information about EPM and the free download are available on CodePlex, under Downloads: <http://www.codeplex.com/EPMFramework>

# Summary and Conclusions

Considering the only tool database administrators had in the past to control the setup of servers and databases was a paper-based policy manual, discovering policy violations and who was responsible for them was a time-consuming task. Furthermore, it was often difficult to discover why a policy had been violated, and even more difficult to determine who violated it. Making sure that a large number of servers were all set up according to standards was a major headache and a waste of time.

The introduction of Policy-Based Management in SQL Server 2008 solves this problem and is proving to be a significant time saver for DBAs and their teams. Through PBM, Central Management Server, and free CodePlex projects like EPM, it is now possible to define rules around server configurations and let SQL Server reason over these policies to enforce the rules and notify you when servers are in violation.

# Appendix: Policy-Based Management Facets

**Note**: All facets support On Demand evaluation, so that mode is not listed in the Evaluation Modes permitted column.

**About the data types**: The data types are intended to tell you what choices will be available to you when defining a condition for that property. For example, if a particular property is listed as “bit”, your choices will be the equal and not equal operators and the values true and false. The data types listed may not in fact be the way the value is represented internally. There are some properties for which I was unable to determine the data type. Most of these are character strings of some sort but I was unable to determine their length.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Facet Name** | | **Evaluation Modes permitted** | **Target(s)** | **Properties** | **Datatypes** |
| Application Role | | On change: Prevent  On Change: Log  On Schedule | Application role | CreateDate  DateLastModified  ID  DefaultSchema  Name | datetime  datetime  int  sysname  sysname |
| Asymmetric Key | | On Change: Prevent  On Change: Log  On Schedule | Asymmetric key | ID  KeyEncryptionAlgorithm  KeyLength  Owner  PrivateKeyEncryptionType  PublicKey  SID  Thumbprint  ProviderName  Name | int  char(4)  int  sysname  char(2)  varbinary(max)  varbinary(85)  varbinary(32)  nvarchar(120)  sysname |
| Audit | | On Schedule | Audit | CreateDate  DateLastModified  DestinationType  Enabled  FileName  FilePath  Guid  ID  MaximumFileSize  MaximumFileSizeUnit  MaximumRolloverFiles  OnFailure  QueueDelay  ReserveDiskSpace  Name | datetime  datetime  char(2)  bit  nvarchar(260)  nvarchar(260)  uniqueidentifier  int  bigint  int  tinyint  int  int  sysname |
| Backup Device | | On Schedule | Backup device | BackupDeviceType  PhysicalLocation  SkipTapeLabel  Name | tinyint  nvarchar(260)  bit  sysname |
| Broker Priority | | On Schedule | Broker priority | ContractName  ID  LocalServiceName  PriorityLevel  RemoteServiceName  Name | sysname  int  sysname  tinyint  sysname  sysname |
| Broker Service | | On Schedule | Broker service | ID  IsSystemObject  Owner  QueueName  QueueSchema  Name | int  bit  sysname  sysname  sysname |
| Certificate | | On Schedule | Certificate | ActiveForServiceBrokerDialog  ExpirationDate  ID  Issuer  Owner  PrivateKeyEncryptionType  Serial  Sid  StartDate  Subject  Thumbprint  LastBackupDate  Name | bit  datetime  int  nvarchar(442)  sysname  char(2)  nvarchar(64)  varbinary(85)  datetime  nvarchar(4000)  varbinary(32)  datetime  sysname |
| Credential | | On Schedule | Credential | CreateDate  DateLastModified  ID  Identity  MappedClassType  ProviderName  Name | datetime  datetime  int  nvarchar(4000)  nvarchar(100)  sysname |
| Cryptographic Provider | | On Schedule | Cryptographic provider | AsymmetricKeyExportable  AsymmetricKeyImportable  AsymmetricKeyPersistable  AsymmetricKeySupported  AuthenticationType  DllPath  Enabled  ID  ProviderGuid  SymmetricKeyExportable  SymmetricKeyImportable  SymmetricKeyPersistable  SymmetricKeySupported  Name | bit  bit  bit  bit  char(4)  nvarchar(512)  bit  int  uniqueidentifier  bit  bit  bit  bit  sysname |
| Data File | | On Schedule | Data file | AvailableSpace  FileName  Growth  GrowthType  ID  IsPrimaryFile  MaxSize  Size  UsedSpace  IsOffline  IsReadOnly  IsReadOnlyMedia  IsSparse  Name | float  nvarchar(260)  float  nvarchar(10)  int  bit  float  float  float  bit  bit  bit  bit  sysname |
| Database | | On Schedule | Database | ActiveConnections  AutoClose  AutoShrink  CompatibilityLevel  CreateDate  DataSpaceUsage  DboLogin  DefaultFileGroup  DefaultSchema  ID  IndexSpaceUsage  IsAccessible  IsDbAccessAdmin  IsDbBackupOperator  IsDbDatareader  IsDbDatawriter  IsDbDdlAdmin  IsDbDenyDatareader  IsDbDenyDatawriter  IsDbOwner  IsDbSecurityAdmin  IsFullTextEnabled  IsSystemObject  LastBackupDate  LastDifferentialBackupDate  LastLogBackupDate  Owner  PrimaryFilePath  ReplicationOptions  Size  SpaceAvailable  Status  UserName  AnsiNullDefault  AnsiNullsEnabled  AnsiPaddingEnabled  AnsiWarningsEnabled  ArithmeticAbortEnabled  AutoCreateStatisticsEnabled  AutoUpdateStatisticsEnabled  CaseSensitive  CloseCursorsOnCommitEnabled  Collation  ConcatenateNullYieldsNull  DatabaseOwnershipChaining  IsUpdateable  LocalCursorsDefault  NumericRoundAbortEnabled  PageVerify  QuotedIdentifiersEnabled  ReadOnly  RecoveryModel  RecursiveTriggersEnabled  UserAccess  Version  AutoUpdateStatisticsAsync  BrokerEnabled  DatabaseGuid  DatabaseSnapshotBaseName  DateCorrelationOptimization  DefaultFullTextCatalog  IsDatabaseSnapshot  IsDatabaseSnapshotBase  IsMailHost  IsMirroringEnabled  IsParameterizationForced  IsReadCommittedSnapshotOn  IsVarDecimalStorageFormatEnabled  LogReuseWaitStatus  MirroringFailoverLogSequenceNumber  MirroringID  MirroringPartner  MirroringPartnerInstance  MirroringRedoQueueMaxSize  MirroringRoleSequence  MirroringSafetyLevel  MirroringSafetySequence  MirroringStatus  MirroringTimeout  MirroringWitness  MirroringWitnessStatus  RecoveryForkGuid (uniqueidentifier)  ServiceBrokerGuid  SnapshotIsolationState  Trustworthy  ChangeTrackingAutoCleanUp  ChangeTrackingEnabled  ChangeTrackingRetentionPeriod  ChangeTrackingRetentionPeriodUnits  DefaultFileStreamFileGroup  EncryptionEnabled  HonorBrokerPriority  IsManagementDataWarehouse  Name | int  bit  bit  tinyint  datetime  float  bit  int  int  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  datetime  datetime  datetime  sysname  float  float  nvarchar(60)  sysname  bit  bit  bit  bit  bit  bit  bit  bit  bit  sysname  bit  bit  bit  bit  bit  nvarchar(60)  bit  bit  nvarchar(60)  bit  nvarchar(60)  int  bit  bit  uniqueidentifier  sysname  bit  sysname  bit  bit  bit  bit  bit  bit  bit  nvarchar(60)  uniqueidentifier  int  int  int  int  uniqueidentifier  uniqueidentifier  bit  bit  bit  int  sysname  bit  bit  bit  sysname |
| Database Audit Specification | | On Schedule | Database audit specification | AuditName  CreateDate  DateLastModified  Enabled  Guid  ID  Name | sysname  datetime  datetime  bit  uniqueidentifier  int  sysname |
| Database DDL Trigger | | On Schedule | Database DDL trigger | AnsiNullsStatus  AssemblyName  BodyStartIndex  ClassName  CreateDate  DateLastModified  ExecutionContext  ExecutionContextUser  ID  ImplementationType  IsEnabled  IsEncrypted  IsSystemObject  MethodName  NotForReplication  QuotedIdentifierStatus  Text  Name | bit  sysname  int  datetime  datetime  int  bit  bit  bit  bit  bit  sysname |
| Database Maintenance | | On Schedule | Database | RecoveryModel  ReadOnly  PageVerify  Status  LastBackupDate  LastLogBackupDate  DataAndBackupOnSeparateLogicalVolumes | nvarchar(60)  bit  nvarchar(60)  nvarchar(60)  datetime  datetime  bit |
| Database Option | | On Change: Log  On Schedule | Database | AnsiNullDefault  AnsiNullsEnabled  AnsiPaddingEnabled  AnsiWarningsEnabled  ArithmeticAbortEnabled  AutoClose  AutoCreateStatisticsEnabled  AutoShrink  AutoUpdateStatisticsAsync  AutoUpdateStatisticsEnabled  BrokerEnabled  ChangeTrackingAutoCleanUp  ChangeTrackingEnabled  ChangeTrackingRetentionPeriod  ChangeTrackingRetentionPeriodUnits  CloseCursorsOnCommitEnabled  Collation  CompatibilityLevel  ConcatenateNullYieldsNull  CreateDate  DatabaseOwnershipChaining  DatabaseSnapshotBaseName  DateCorrelationOptimization  DefaultFileGroup  DefaultFileStreamFileGroup  EncryptionEnabled  HonorBrokerPriority  ID  IsParameterizationForced  IsReadCommittedSnapshotOn  IsSystemObject  IsUpdateable  LocalCursorsDefault  Name  Owner  NumericRoundAbortEnabled  MirroringTimeout  PageVerify  PrimaryFilePath  QuotedIdentifiersEnabled  ReadOnly  RecoveryModel  RecursiveTriggersEnabled  Trustworthy  UserAccess | bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  bit  int  bit  tinyint  bit  datetime  bit  sysname  bit  sysname  sysname  bit  bit  int  bit  bit  bit  bit  bit  sysname  sysname  bit  int  nvarchar(60)  bit  bit  nvarchar(60)  bit  bit  nvarchar(60) |
| Database Performance | | On Schedule | Database | AutoClose  AutoShrink  Size  DataAndLogFilesOnSeparateLogicalVolumes  CollationMatchesModelOrMaster  IsSystemObject  Status | bit  bit  float  bit  bit  bit |
| Database Role | | On Change: Prevent  On Change: Log  On Schedule | Database role | CreateDate  DateLastModified  ID  IsFixedRole  Owner  Name | datetime  datetime  int  bit  sysname  sysname |
| Database Security | | On Schedule | Database | Trustworthy  IsOwnerSysadmin | bit  bit |
| Default | | On Schedule | Default | CreateDate  ID  Schema  Name | datetime  int  sysname  sysname |
| Endpoint | | On Schedule | Endpoint | EndpointState  EndpointType  ID  IsAdminEndpoint  IsSystemObject  Owner ProtocolType  Name | nvarchar(60)  nvarchar(60)  int bit bit sysname nvarchar(60) sysname |
| File Group | | On Schedule | File group | ID (System.Int32)  IsDefault  ReadOnly  Size  IsFileStream  Name | Int bit bit float bit sysname |
| Full Text Catalog | | On Schedule | Full text catalog | ErrorLogSize  FullTextIndexSize  HasFullTextIndexedTables  ID  ItemCount  PopulationCompletionDate  PopulationStatus  RootPath  UniqueKeyCount  FileGroup  IsAccentSensitive  IsDefault  Owner  Name | int int bit int int datetime  nvarchar(260) int sysname bit bit sysname sysname |
| Full Text Index | | On Schedule | Full text index | CatalogName  ChangeTracking  IsEnabled  PopulationStatus UniqueIndexName  DocumentsProcessed  ItemCount  NumberOfFailures  PendingChanges  FilegroupName  StopListName  StopListOption | sysname  bit  sysname  int  int  int  int  sysname |
| Full Text Stop List | | On Schedule | Full text stop list | ID  Owner (System.String)  Name | int  sysname |
| Index | | On Schedule | Index | DisallowPageLocks  DisallowRowLocks  FileGroup  FillFactor  IgnoreDuplicateKeys  IndexKeyType  IsClustered  IsFullTextKey  IsSystemNamed  IsSystemObject  IsUnique  NoAutomaticRecomputation  PadIndex  SpaceUsed  IsDisabled  IsPartitioned  IsXmlIndex  ParentXmlIndex  PartitionScheme  SecondaryXmlIndexType  BoundingBoxXMax  BoundingBoxXMin  BoundingBoxYMax  BoundingBoxYMin  CellsPerObject  FileStreamFileGroup  FileStreamPartitionScheme  FilterDefinition (System.String)  HasCompressedPartitions (System.Boolean)  HasFilter  IsSpatialIndex  Level1Grid  Level2Grid  Level3Grid  Level4Grid  SpatialIndexType  Name | bit  bit  sysname  tinyint  bit  bit  bit  bit  bit  bit  bit  bit  bit  float  bit  bit  bit  sysname  sysname  float  float  float  float  int  sysname  sysname  nvarchar(max)  bit  bit  bit  sysname |
| Linked Server | | On Schedule | Linked server | Catalog  DataAccess  DataSource  DistPublisher  Distributor  ID  Location  ProductName  ProviderName  Publisher  Rpc  RpcOut  Subscriber  CollationName  ConnectTimeout  LazySchemaValidation  QueryTimeout  UseRemoteCollation  DateLastModified  IsPromotionofDistributedTransactionsForRPCEnabled  Name (System.String) i | bit  bit  bit  int  bit  bit  bit  bit  int  bit  int  bit  datetime  bit  bit |
| Log File | | On Schedule | Log file | FileName  Growth  GrowthType  ID  MaxSize  Size  UsedSpace  IsOffline IsReadOnly  IsReadOnlyMedia  IsSparse  Name | float  int  float  float  float  bit  bit  bit  bit  sysname |
| Login | | On Schedule | Login | CreateDate  DateLastModified  DefaultDatabase  DenyWindowsLogin  HasAccess  IsSystemObject  Language  LanguageAlias  LoginType  Sid  WindowsLoginAccessType  AsymmetricKey  Certificate  Credential  ID  IsDisabled  IsLocked  IsPasswordExpired  MustChangePassword  PasswordExpirationEnabled  PasswordPolicyEnforced  Name | datetime  datetime  sysname  bit  bit  bit  sysname  nvarchar(60)  varbinary(85)  sysname  sysname  sysname  int  bit  bit  bit  bit  bit  bit  sysname |
| Login Options | | On Change: Prevent  On Change: Log  On Schedule | Login | AsymmetricKey  Certificate  CreateDate  Credential  DefaultDatabase  ID  IsDisabled  IsLocked  IsSystemObject  Language  LanguageAlias  MustChangePassword  Name (System.String)  PasswordExpirationEnabled  PasswordPolicyEnforced | sysname  sysname  datetime  sysname  sysname  int  bit  bit  bit  sysname  bit  sysname  bit  bit |
| Message Type | | On Schedule | Message type | ID  IsSystemObject  MessageTypeValidation  ValidationXmlSchemaCollection  ValidationXmlSchemaCollectionSchema  Name | int  bit  sysname  sysname  sysname |
| Multipart Name | | On Change Prevent  On Change: Log  On Schedule | Function  Procedure  Synonym  Table  Type  View | Name  Schema | sysname  sysname |
| Name | | On Schedule | Application role  Asymmetric key  Certificate  Database role  Default  Index  Rule  Schema  SQL Assembly  Stored procedure  Symmetric key  Synonym  Table  Trigger  User  User defined function  User defined type  View  XML Schema Collection | Name | sysname |
| Partition Function | | On Schedule | Partition function | CreateDate  ID  NumberOfPartitions  RangeType  Name | datetime  int  int  sysname |
| Partition Scheme | | On Schedule | Partition scheme | ID  NextUsedFileGroup  PartitionFunction  Name | int  sysname  sysname  sysname |
| Plan Guide | | On Schedule | Plan guide | Hints  ID  IsDisabled  Parameters  ScopeBatch  ScopeObjectName  ScopeSchemaName  ScopeType  Statement  Name | Int  Bit  sysname |
| Remote Service Binding | On Schedule | | Remote service binding | CertificateUser  IsAnonymous  Owner  RemoteService  Name | sysname  bit  sysname  sysname  sysname |
| Resource Governor | On Schedule | | Resource governor | ClassifierFunction  Enabled  ReconfigurePending | sysname  bit  bit |
| Resource Pool | On Change: Prevent  On Change: Log  On Schedule | | Resource pool | ID  IsSystemObject  MaximumCpuPercentage  MaximumMemoryPercentage  MinimumCpuPercentage  MinimumMemoryPercentage  Name | int  bit  int  int  int  int  sysname |
| Rule | On Schedule | | Rule | CreateDate  ID  DateLastModified  Schema  Name | datetime  int  datetime  sysname  sysname |
| Schema | On Change: Prevent  On Change: Log  On Schedule | | Schema | ID  Owner  Name | int  sysname  sysname |
| Server | On Schedule | | Server | AuditLevel  BackupDirectory  BuildNumber  DefaultFile  DefaultLog  ErrorLogPath  InstallDataDirectory  IsCaseSensitive  IsFullTextInstalled  Language  MailProfile  MasterDBLogPath  MasterDBPath  MaxPrecision  NumberOfLogFiles  OSVersion  PerfMonMode  PhysicalMemory  Platform  Processors  Product  RootDirectory  ServiceName  VersionMajor  VersionMinor  Collation  Edition  EngineEdition  InstanceName  IsClustered  IsSingleUser  NetName  ProductLevel  ServerType  Status  TapeLoadWaitTime  VersionString  BrowserServiceAccount  BrowserStartMode  BuildClrVersionString  CollationID  ComparisonStyle  ComputerNamePhysicalNetBIOS  InstallSharedDirectory  NamedPipesEnabled  ResourceLastUpdateDateTime  ResourceVersionString  ServiceAccount  ServiceInstanceId  ServiceStartMode  SqlCharSet  SqlCharSetName  SqlSortOrder  SqlSortOrderName  TcpEnabled  FilestreamLevel  FilestreamShareName  SqlDomainGroup | Int  Bit  Bit  Tinyint  Int  Int  Int  Int  Int  sysname  sysname  sysname  sysname  Bit  Bit  sysname  sysname  int  sysname  bit  datetime  smallint  smallint  bit |
| Server Audit | On Schedule | | Server audit | DefaultTraceEnabled  C2AuditTracingEnabled  LoginAuditLevel | bit  bit |
| Server Audit Specification | On Schedule | | Server audit specification | AuditName  CreateDate  DateLastModified  Enabled  Guid  ID  Name | sysname  datetime  datetime  bit  uniqueidentifier  int  sysname |
| Server Configuration | On Change: Log  On Schedule | | Server | AdHocRemoteQueriesEnabled  AffinityMask  Affinity64Mask  AffinityIOMask  Affinity64IOMask  AgentXPsEnabled  AllowUpdates  AweEnabled  BlockedProcessThreshold  C2AuditTracingEnabled  ClrIntegrationEnabled  CommonCriteriaComplianceEnabled  CostThresholdForParallelism  CrossDBOwnershipChainingEnabled  CursorThreshold  DatabaseMailEnabled  DefaultTraceEnabled  DefaultFullTextLanguage  DefaultLanguage  DisallowResultsFromTriggers ( FillFactor  FullTextCrawlBandwidthMin  FullTextCrawlBandwidthMax  FullTextNotifyBandwidthMin  FullTextNotifyBandwidthMax  FullTextCrawlRangeMax  InDoubtTransactionResolution  IndexCreateMemory  LightweightPoolingEnabled  DynamicLocks  MaxDegreeOfParallelism  MaxServerMemory  MaxWorkerThreads  MediaRetention  MinMemoryPerQuery  MinServerMemory  NestedTriggersEnabled  NetworkPacketSize  OleAutomationEnabled  OpenObjects  PrecomputeRank  PriorityBoost  ProtocolHandlerTimeout  QueryGovernorCostLimit  QueryWait  RecoveryInterval  RemoteAccessEnabled  RemoteDacEnabled  RemoteLoginTimeout  RemoteProcTransEnabled  RemoteQueryTimeout  ReplicationMaxTextSize  ReplicationXPsEnabled  ScanForStartupProcedures  ServerTriggerRecursionEnabled  SetWorkingSetSize  ShowAdvancedOptions  SmoAndDmoXPsEnabled  SqlMailEnabled (System.Boolean)  TransformNoiseWords  TwoDigitYearCutoff  UserConnections  UserInstanceTimeout  UserInstancesEnabled  UserOptions  WebAssistantEnabled  XPCmdShellEnabled  DefaultBackupCompressionEnabled  ExtensibleKeyManagementEnabled  FilestreamAccessLevel  OptimizeAdhocWorkloads | bit  int  int  bit  bit  bit  int  bit  bit  bit  bit  bit  int  bit  int  bit  bit  int  int  bit  int  int  int  int  int  int  int  bit  int  int  int  int  int  int  int  bit  int  int  int  bit  bit  int  int  int  int  bit  bit  int  bit  int  int  bit  bit  bit  bit  bit  bit  bit  bit  int  int  int  int  int  bit  bit  bit  bit  tinyint  bit |
| Server DDL Trigger | On Schedule | | Server DDL trigger | AnsiNullsStatus  AssemblyName  BodyStartIndex  ClassName  CreateDate  DateLastModified  ExecutionContext  ExecutionContextLogin  ID  ImplementationType  IsEnabled  IsEncrypted  IsSystemObject  MethodName  QuotedIdentifierStatus  Text (System.String)  Name | bit  bit  int  sysname  datetime  datetime  sysname  int  bit  bit  bit  sysname  bit  varbinary(max)  sysname |
| Server Information | On Schedule | | Server | Collation  Edition  ErrorLogPath  IsCaseSensitive  IsClustered  IsFullTextInstalled  IsSingleUser  Language  MasterDBLogPath  MasterDBPath  MaxPrecision  NetName  OSVersion  PhysicalMemory  Platform  Processors  Product  ProductLevel  RootDirectory  VersionString  EngineEdition  VersionMajor  VersionMinor  BuildClrVersionString  BuildNumber  CollationID  ComparisonStyle  ComputerNamePhysicalNetBIOS  ResourceLastUpdateDateTime  ResourceVersionString  SqlCharSet  SqlCharSetName  SqlSortOrder  SqlSortOrderName (System.String) | sysname  sysname  bit  bit  bit  bit  sysname  tinyint  sysname  string  int  sysname  int  int  sysname  int  int  int  sysname  datetime  sysname  tinyint  sysname  tinyint  sysname |
| Server Installation | On Schedule | | Server | ServiceName  EngineServiceAccount  ServiceStartMode  InstanceName  ServiceInstanceIdSuffix  FilestreamLevel  FilestreamShareName  UserInstancesEnabled  Collation  SqlDomainGroup  WindowsUsersAndGroupsInSysadminRole  LoginMode  InstallDataDirectory  BackupDirectory  DefaultFile  DefaultLog  TempdbPrimaryFilePath  TempdbLogPath  AgentStartMode  AgentServiceAccount  AgentDomainGroup  NamedPipesEnabled  TcpEnabled  InstallSharedDirectory  BrowserStartMode  BrowserServiceAccount | sysname  tinyint  sysname  sysname  bit  bit |
| Server Performance | On Schedule | | Server | Affinitymask  Affinity64mask  Affinityiomask  Affinity64iomask  BlockedProcessThreshold  DynamicLocks  Lightweightpoolingenabled  MaxDegreeofParallelism  CostThresholdforParallelism  MaxWorkerThreads  NetworkPacketSize  Openobjects | int  int  int  int  int  int  bit  bit  int  int  int  int |
| Server Security | On Schedule | | Server | PublicServerRoleIsGrantedPermissions  LoginMode  XPCmdShellEnabled  CrossDBOwnershipChainingEnabled  CommonCriteriaComplianceEnabled  CmdExecRightsForSystemAdminsOnly  ProxyAccountIsGrantedToPublicRole  ReplaceAlertTokensEnabled  ProxyAccountEnabled | bit  bit  bit  bit  bit  bit  bit  bit |
| Server Settings | On Schedule | | Server | AuditLevel  BackupDirectory  DefaultFile  DefaultLog  LoginMode  MailProfile  NumberOfLogFiles  PerfMonMode  TapeLoadWaitTime | sysname  int  int |
| Service Contract | On Schedule | | Service contract | ID  IsSystemObject  Owner  Name | Int  Bit  Sysname  sysname |
| Service Queue | On Schedule | | Service queue | ActivationExecutionContext  CreateDate  DateLastModified  ExecutionContextPrincipal  FileGroup  ID  IsActivationEnabled  IsEnqueueEnabled  IsRetentionEnabled  IsSystemObject  MaxReaders  ProcedureDatabase  ProcedureName  ProcedureSchema  RowCount  Schema  Name | sysname  datetime  datetime  sysname  sysname  int  bit  bit  bit  bit  smallint  sysname  sysname  sysname  bigint  sysname  sysname |
| Service Route | On Schedule | | Service route | Address  BrokerInstance  ExpirationDate  ID  MirrorAddress   Owner  RemoteService  Name | datetime  int  sysname  sysname |
| Statistic | On Schedule | | Statistic | FileGroup  ID  IsAutoCreated  IsFromIndexCreation  LastUpdated  NoAutomaticRecomputation  FilterDefinition  HasFilter  Name | sysname  int  bit  bit  datetime  bit  varbinary(max)  bit  sysname |
| Stored Procedure | On Change: Prevent  On Change: Log  On Schedule | | Stored procedure | AnsiNullsStatus  CreateDate  ForReplication  ID  ImplementationType  IsEncrypted  IsSystemObject  Owner  QuotedIdentifierStatus  Recompile  Startup  AssemblyName  ClassName  ExecutionContext  ExecutionContextPrincipal  IsSchemaOwned  MethodName  Schema  Name | bit  datetime  bit  int  bit  bit  sysname  bit  bit  bit  sysname  sysname  sysname  bit  sysname  sysname  sysname |
| Surface Area | On Change: Log  On Schedule | | Server | AdHocRemoteQueriesEnabled  DatabaseMailEnabled  ClrIntegrationEnabled  OleAutomationEnabled  RemoteDacEnabled (System.Boolean)  SqlMailEnabled (System.Boolean)  WebAssistantEnabled  XPCmdShellEnabled  ServiceBrokerEndpointActive  SoapEndpointsEnabled | bit  bit  bit  bit  bit  bit  bit  bit  bit  bit |
| Surface Area for AS |  | | Analysis services | AdHocDataMiningQueriesEnabled  LinkedObjectsLinksToOtherInstancesEnabled  LinkedObjectsLinksFromOtherInstancesEnabled  UserDefinedFunctionsEnabled  ListenOnlyOnLocalConnections | bit  bit  bit  bit  bit |
| Surface Area for RS |  | | Reporting services | WebServiceAndHTTPAccessEnabled  ScheduleEventsAndReportDeliveryEnabled  ReportManagerEnabled | bit  bit  bit |
| Symmetric Key | On Schedule | | Symmetric key | CreateDate  DateLastModified  EncryptionAlgorithm  ID  IsOpen  KeyGuid  KeyLength  Owner  ProviderName  Name | datetime  datetime  nvarchar(60)  int  bit  uniqueidentifier  int  sysname  nvarchar(120)  sysname |
| Synonym | On Schedule | | Synonym | BaseDatabase  BaseObject  BaseSchema  BaseServer  BaseType  CreateDate  DateLastModified  ID  IsSchemaOwned  Owner  Schema  Name | sysname  sysname  sysname  sysname  sysname  datetime  datetime  int  bit  sysname  sysname  sysname |
| Table | On Schedule | | Table | CreateDate  DataSpaceUsed  FakeSystemTable  FileGroup  HasClusteredIndex  ID  IndexSpaceUsed  IsSystemObject  Owner  Replicated  RowCount  TextFileGroup  AnsiNullsStatus  HasAfterTrigger  HasDeleteTrigger  HasIndex  HasInsertTrigger  HasInsteadOfTrigger  HasUpdateTrigger  IsIndexable  QuotedIdentifierStatus  DateLastModified  IsPartitioned  IsSchemaOwned  IsVarDecimalStorageFormatEnabled  PartitionScheme  ChangeTrackingEnabled  FileStreamFileGroup  FileStreamPartitionScheme  HasCompressedPartitions  LockEscalation  TrackColumnsUpdatedEnabled  Schema  Name | datetime  float  bit  sysname  bit  int  float  bit  sysname  bit  bigint  sysname  bit  bit  bit  bit  bit  bit  bit  bit  bit  datetime  bit  bit  bit  sysname  bit  sysname  sysname  bit  bit  sysname  sysname |
| Table Options | On Change: Prevent  On Change: Log  On Schedule | | Table | AnsiNullsStatus  ChangeTrackingEnabled  CreateDate  FakeSystemTable  ID  IsSchemaOwned  IsSystemObject  LockEscalation  Name  Owner  QuotedIdentifierStatus  Replicated  Schema  TrackColumnsUpdatedEnabled | bit  bit  datetime  bit  int  bit  bit  bit  sysname  sysname  bit  bit  sysname  bit |
| Trigger | On Schedule | | Trigger | AnsiNullsStatus  CreateDate  Delete  DeleteOrder  ID  ImplementationType  Insert  InsertOrder  InsteadOf (System.Boolean)  IsEnabled  IsEncrypted (System.Boolean)  IsSystemObject  NotForReplication  QuotedIdentifierStatus  Update  UpdateOrder  AssemblyName  ClassName  DateLastModified  ExecutionContext  ExecutionContextPrincipal  MethodName  Name | bit  bit  bit  int  bit  bit  bit  bit  bit  bit  bit  bit  sysname  sysname  datetime  sysname  sysname  sysname |
| User | On Schedule | | User | CreateDate  DateLastModified  HasDBAccess  ID  IsSystemObject  Login  LoginType  Sid  UserType  AsymmetricKey  Certificate  DefaultSchema  Name | datetime  datetime  bit  int  bit  sysname  varbinary(85)  sysname  sysname  sysname  sysname |
| User Defined Aggregate | On Schedule | | User defined aggregate | AssemblyName  ClassName  CreateDate  DateLastModified  ID  IsSchemaOwned  Owner  Schema  Name | sysname  sysname  datetime  datetime  int  bit  sysname  sysname  sysname |
| User Defined Data Type | On Schedule | | User defined data type | AllowIdentity  Default  DefaultSchema  ID  Length  MaxLength  Nullable  NumericPrecision  NumericScale  Owner  Rule  RuleSchema  SystemType  VariableLength  Collation  IsSchemaOwned  Schema  Name (System.String) | bit  sysname  sysname  int  int  int  bit  int  int  sysname  sysname  sysname  sysname  bit  sysname  bit  sysname  sysname |
| User Defined Function | On Change: Prevent  On Change: Log  On Schedule | | User defined function | AnsiNullsStatus  CreateDate  FunctionType  ID  ImplementationType  IsDeterministic  IsEncrypted  IsSchemaBound  IsSystemObject  Owner  QuotedIdentifierStatus  TableVariableName  AssemblyName  ClassName  ExecutionContext  ExecutionContextPrincipal  IsSchemaOwned  MethodName  ReturnsNullOnNullInput  Schema  Name | bit  datetime  int  bit  bit  bit  bit  sysname  bit  sysname  sysname  sysname  sysname  bit  sysname  bit  sysname  sysname |
| User Defined Table Type | On Schedule | | User defined table type | Collation  CreateDate  DateLastModified  ID  IsSchemaOwned  IsUserDefined  MaxLength  Nullable  Owner  Schema  Name | sysname  datetime  datetime  int  bit  bit  smallint  bit  sysname  sysname  sysname |
| User Defined Type | On Schedule | | User defined type | AssemblyName  BinaryTypeIdentifier  ClassName  Collation  ID  IsBinaryOrdered  IsComVisible  IsFixedLength  IsNullable  IsSchemaOwned  MaxLength  NumericPrecision  NumericScale  Owner  UserDefinedTypeFormat  Schema  Name | sysname  varbinary  sysname  sysname  int  bit  bit  bit  bit  bit  int  int  int  sysname  sysname  sysname |
| User Options | On Change: Prevent  On Change: Log  On Schedule | | Asymmetric key user  Certificate user  Group user  SQL user  Windows user | AsymmetricKey  Certificate  CreateDate  DefaultSchema  ID  IsSystemObject  Login  LoginType  Name  Sid  UserType | sysname  sysname  sysname  sysname  int  bit  bit  sysname  varbinary(85) |
| View | On Schedule | | View | AnsiNullsStatus  CreateDate  HasColumnSpecification  ID  IsEncrypted  IsSchemaBound  IsSystemObject  Owner  QuotedIdentifierStatus  HasAfterTrigger  HasDeleteTrigger  HasIndex  HasInsertTrigger  HasInsteadOfTrigger  HasUpdateTrigger  IsIndexable  DateLastModified  IsSchemaOwned  ReturnsViewMetadata  Schema  Name | bit  datetime  bit  int  bit  bit  bit  sysname  bit  bit  bit  bit  bit  bit  bit  bit  datetime  bit  bit  sysname  sysname |
| View Options | On Change: Prevent  On Change: Log  On Schedule | | View | AnsiNullsStatus  CreateDate  ID  IsEncrypted  IsSchemaBound  IsSchemaOwned  IsSystemObject  Name  Owner  Schema  QuotedIdentifierStatus  ReturnsViewMetadata | bit  datetime  int  bit  bit  bit  bit  sysname  sysname  sysname  bit  bit |
| Workload Group | On Change: Prevent  On Change: Log  On Schedule | | Workload group | GroupMaximumRequests  ID  Importance  IsSystemObject  MaximumDegreeOfParallelism  RequestMaximumCpuTimeInSeconds  RequestMaximumMemoryGrantPercentage  RequestMemoryGrantTimeoutInSeconds  Name | int  int  bit  int  int  int  int  sysname |
| Xml Schema Collection | On Schedule | | XML schema collection | CreateDate  DateLastModified  ID  Text  Schema  Name | datetime  datetime  int  nvarchar(max)  sysname  sysname |

Sources: in preparing this chart, I used the list of facets and properties generated by Jens Suessmeyer. The code for generating this list can be found at <http://blogs.msdn.com/sqlpbm/>, and the list can be found at <http://blogs.msdn.com/jenss/archive/2009/04/18/getting-a-list-of-all-facets-and-its-properties.aspx>. I also used the chart of evaluation modes and targets provided by Dan Jones at <http://blogs.msdn.com/sqlpbm/archive/2008/05/24/facets.aspx>. I also used the data declarations for the catalog views to determine the data types.

# About the Author

Sharon Dooley (MCP, former SQL Server MVP) has worked with SQL Server since its first release. Her major areas of specialization are database design and performance tuning, though she also does a lot of administration. She has a consulting practice and teaches and writes courses for Learning Tree International, where she is SQL Server Curriculum Manager. Ms. Dooley is the author of *SQL Server 7.0 Essential Reference* (New Riders), the lead author of *Professional SQL Server 6.5 Administration* (Wrox). Sharon is an active member of PASS (The Professional Organization for SQL Server) and has presented at numerous conferences.

**For more information:**

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20Policy-Based%20Management).