Administer Servers by Using Policy-Based Management

08/12/2016 • 5 minutes to read • Contributors

In this article

Three Policy-Based Management components
Terms
Links to specific tasks
Examples

THIS TOPIC APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

Policy-Based Management is a policy based system for managing one or more instances of SQL Server. Use is to create conditions that contain condition expressions. Then, create policies that apply the conditions to database target objects.

For example, as the database administrator, you may want to ensure that certain servers do not have Database Mail enabled, so you create a condition and a policy that sets that server option.

> **IMPORTANT!!** Policies can affect how some features work. For example, change data capture and transactional replication both use the systranschemas table, which does not have an index. If you enable a policy that all tables must have an index, enforcing compliance of the policy will cause these features to fail.

Use SQL Server management Studio to create and manage policies, to:

1. Select a Policy-Based Management facet that contains the properties to be configured.
2. Define a condition that specifies the state of a management facet.

3. Define a policy that contains the condition, additional conditions that filter the target sets, and the evaluation mode.

4. Check whether an instance of SQL Server is in compliance with the policy.

   For failed policies, Object Explorer indicates a critical health warning as a red icon next to the target and the nodes that are higher in the Object Explorer tree.

> **NOTE:** When the system computes the object set for a policy, by default the system objects are excluded. For example, if the object set of the policy refers to all tables, the policy will not apply to system tables. If users want to evaluate a policy against system objects, they can explicitly add system objects to the object set. However, though all policies are supported for **check on schedule** evaluation mode, for performance reason, not all policies with arbitrary object sets are supported for **check on change** evaluation mode. For more information, see http://blogs.msdn.com/b/sqlpbm/archive/2009/04/13/policy-evaluation-modes.aspx

Three Policy-Based Management components

Policy-Based Management has three components:

- Policy management. Policy administrators create policies.
- Explicit administration. Administrators select one or more managed targets and explicitly check that the targets comply with a specific policy, or explicitly make the targets comply with a policy.
- Evaluation modes. There are four evaluation modes; three can be automated:
  - **On demand**. This mode evaluates the policy when directly specified by the user.
  - **On change: prevent**. This automated mode uses DDL triggers to prevent policy violations.

    > **IMPORTANT!** If the nested triggers server configuration option is disabled, **On change: prevent** will not work correctly. Policy-Based Management relies on DDL triggers to detect and roll back DDL operations that do not comply with policies that use this evaluation mode. Removing the Policy-Based Management DDL triggers or disabling nest triggers, will cause this evaluation mode to fail or perform unexpectedly.

  - **On change: log only**. This automated mode uses event notification to evaluate a policy when a relevant change is made.
  - **On schedule**. This automated mode uses a SQL Server Agent job to periodically evaluate a policy.

When automated policies are not enabled, Policy-Based Management will not affect system performance.

Terms

Policy-Based Management managed target Entities that are managed by Policy-Based Management, such as an instance of the SQL Server Database Engine, a database, a table, or an index. All targets in a server instance form a target hierarchy. A target set is the set of targets that results from applying a set of target filters to the target hierarchy, for example, all the tables in the database owned by the HumanResources schema.

Policy-Based Management facet A set of logical properties that model the behavior or characteristics for certain types of managed targets. The number and characteristics of the properties are built into the facet and can be added or removed by only the maker of the facet. A target type can implement one or more management facets, and a management facet can be implemented by one or more target types. Some properties of a facet can only apply to a specific version..

Policy-Based Management condition
A Boolean expression that specifies a set of allowed states of a Policy-Based Management managed target with regard to a management facet. SQL Server tries to observe collations when evaluating a condition. When SQL Server collations do not exactly match Windows collations, test your condition to determine how the algorithm resolves conflicts.

Policy-Based Management policy
A Policy-Based Management condition and the expected behavior, for example, evaluation mode, target filters, and schedule. A policy can contain only one condition. Policies can be enabled or disabled. Policies are stored in the msdb database.

Policy-Based Management policy category
A user-defined category to help manage policies. Users can classify policies into different policy categories. A policy belongs to one and only one policy category. Policy categories apply to databases and servers. At the database level, the following conditions apply:

- Database owners can subscribe a database to a set of policy categories.
- Only policies from its subscribed categories can govern a database.
- All databases implicitly subscribe to the default policy category.

  At the server level, policy categories can be applied to all databases.

  Effective policy
  The effective policies of a target are those policies that govern this target. A policy is effective with regard to a target only if all the following conditions are satisfied:

- The policy is enabled.
- The target belongs to the target set of the policy.
- The target or one of the targets ancestors subscribes to the policy group that contains this policy.

Links to specific tasks

- Store Policy-Based Management policies.|
- Configure Alerts to Notify Policy Administrators of Policy Failures
- Create a New Policy-Based Management Condition
- Delete a Policy-Based Management Condition
- View or Modify the Properties of a Policy-Based Management Condition|
- Export a Policy-Based Management Policy
- Import a Policy-Based Management Policy|
- Evaluate a Policy-Based Management Policy from an Object
- Work with Policy-Based Management Facets|
- Monitor and Enforce Best Practices Using Policy-Based Management

Examples

- Create the Off By Default Policy
    - Configure a Server to Run the Off By Default Policy ## See also Policy-Based Management Views (Transact-SQL)

> **Note**
>
> The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, **we invite you to read our blog post**.

0 Comments

Sign in

2 people listening

+ Follow

Post comment as...