

Authoring for System Center 2012 – Operations Manager

Microsoft Corporation

Published: January 15, 2013

Authors

Steve Moore and Byron Ricks

Applies To

System Center 2012 – Operations Manager

System Center 2012 Service Pack 1 (SP1) – Operations Manager

Feedback

Send suggestions and comments about this document to [sc2012docs@microsoft.com](mailto:sc2012docs@microsoft.com?subject=System%20Center%202012%20Authoring%20Operations%20%20Manager%20(SP1)%20Documentation%20published%20January,%2015,%202013).

Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious.  No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Bing, Internet Explorer, JScript, SharePoint, Silverlight, SQL Server, Visio, Visual Basic, Visual Studio, Win32, Windows, Windows Intune, Windows PowerShell, and Windows Vista are trademarks of the Microsoft group of companies. Portions of this documentation related to network monitoring are provided by EMC, and for those portions the following copyright notice applies 2010 © EMC Corporation. All rights reserved. All other trademarks are property of their respective owners.

Revision History

|  |  |
| --- | --- |
| Release Date | Changes |
| April 1, 2012 | Original release of this guide. |
| September 10, 2012 | Beta release for System Center 2012 Service Pack 1 (SP1) – Operations Manager |
| January 15, 2013 | Release for System Center 2012 Service Pack 1 (SP1) – Operations Manager |

Contents

[Authoring for System Center 2012 - Operations Manager 6](#_Toc345624438)

[Key Concepts for Authors 7](#_Toc345624439)

[What is an Author? 7](#_Toc345624440)

[Authoring Tools 8](#_Toc345624441)

[Structure of a Management Pack 11](#_Toc345624442)

[Sealed Management Pack Files 14](#_Toc345624443)

[How to Seal a Management Pack File 15](#_Toc345624444)

[Selecting a Management Pack File 18](#_Toc345624445)

[Targets and Objects 20](#_Toc345624446)

[Understanding Classes and Objects 21](#_Toc345624447)

[Selecting a target 25](#_Toc345624448)

[Creating a new target 30](#_Toc345624449)

[Attributes 32](#_Toc345624450)

[Management Pack Templates 37](#_Toc345624451)

[Creating Management Pack Templates 38](#_Toc345624452)

[Watcher Nodes 39](#_Toc345624453)

[.NET Application Performance Monitoring Template 40](#_Toc345624454)

[Before You Begin Monitoring .NET Applications 91](#_Toc345624455)

[How to Configure Monitoring for .NET Applications 94](#_Toc345624456)

[How to Add, Enable, and Disable Namespaces 102](#_Toc345624457)

[How to Add, Edit, and Remove Methods 107](#_Toc345624458)

[How to Add, Edit, and Remove Exception Tracking 108](#_Toc345624459)

[Using Exception Handlers to Define Critical Exceptions 110](#_Toc345624460)

[Client-Side Monitoring with Targeted Groups and Load Balancers 113](#_Toc345624461)

[How to Configure IP Address Exclusion Filters for Client-Side Monitoring 113](#_Toc345624462)

[How to Start Monitoring a New Application 115](#_Toc345624463)

[Authoring Strategies for .NET Application Monitoring 117](#_Toc345624464)

[Application Monitoring Using the Default Settings 121](#_Toc345624465)

[Monitoring SharePoint 2010 Applications in System Center 2012 SP1 122](#_Toc345624466)

[Appendix for .NET Application Monitoring 123](#_Toc345624467)

[Default Entry Points for .NET Application Monitoring 123](#_Toc345624468)

[Default Methods for .NET Application Monitoring 124](#_Toc345624469)

[OLE DB Data Source Template 145](#_Toc345624470)

[Process Monitoring Template 152](#_Toc345624471)

[TCP Port Template 159](#_Toc345624472)

[UNIX or Linux Log File 163](#_Toc345624473)

[UNIX or Linux Process 166](#_Toc345624474)

[Web Application Availability Monitoring Template 170](#_Toc345624475)

[How to Configure Web Application Availability Monitoring 191](#_Toc345624476)

[Monitoring Web Application Availability Tests and Alerts 196](#_Toc345624477)

[Dashboard Views for Web Application Availability Monitoring 197](#_Toc345624478)

[Reporting for Web Application Availability Monitoring 199](#_Toc345624479)

[Web Application Transaction Monitoring Template 201](#_Toc345624480)

[How to Create a Single URL Web Application Monitor 202](#_Toc345624481)

[How to Capture Web Application Recording 203](#_Toc345624482)

[Troubleshooting Web Capture 205](#_Toc345624483)

[How to Edit Settings or Requests in a Web Application 206](#_Toc345624484)

[Web Application Properties 207](#_Toc345624485)

[Web Application Request Properties 209](#_Toc345624486)

[How to Replace Parameters in a URL Request 212](#_Toc345624487)

[Windows Service Template 219](#_Toc345624488)

[Monitors and Rules 224](#_Toc345624489)

[Data Sources 228](#_Toc345624490)

[Expressions 229](#_Toc345624491)

[Alerts 235](#_Toc345624492)

[Event Monitors and Rules 239](#_Toc345624493)

[Windows Events 240](#_Toc345624494)

[Text Logs 245](#_Toc345624495)

[WMI Events 250](#_Toc345624496)

[Syslog Events 256](#_Toc345624497)

[SNMP Events 261](#_Toc345624498)

[Event Monitor Logic 268](#_Toc345624499)

[Repeating Events 269](#_Toc345624500)

[Correlated Events 274](#_Toc345624501)

[Missing Events 276](#_Toc345624502)

[Correlated Missing Events 276](#_Toc345624503)

[Event Monitor Reset 278](#_Toc345624504)

[UNIX/Linux Shell Command Alerts 279](#_Toc345624505)

[Performance Monitors and Rules 283](#_Toc345624506)

[Performance Monitors 283](#_Toc345624507)

[Windows Performance Collection Rules 288](#_Toc345624508)

[WMI Performance 292](#_Toc345624509)

[UNIX/Linux Shell Command Performance Collection Rules 295](#_Toc345624510)

[Script Monitors and Rules 299](#_Toc345624511)

[Script Collection Rules 303](#_Toc345624512)

[Script Monitors 309](#_Toc345624513)

[UNIX/Linux Shell Command Monitors 316](#_Toc345624514)

[Dependency Monitors 321](#_Toc345624515)

[Aggregate Monitors 326](#_Toc345624516)

[Diagnostics and Recoveries 328](#_Toc345624517)

[Tasks 333](#_Toc345624518)

[Console Tasks 334](#_Toc345624519)

[Agent Tasks 339](#_Toc345624520)

[Distributed Applications 345](#_Toc345624521)

[360 .NET Application Monitoring Dashboards in System Center 2012 SP1 349](#_Toc345624522)

[Before You Begin Using 360 .NET Application Monitoring Dashboards in System Center 2012 SP1 352](#_Toc345624523)

[How to Configure 360 .NET Application Monitoring Dashboards in System Center 2012 SP1 353](#_Toc345624524)

[How to Use the 360 .NET Application Monitoring Dashboards in System Center 2012 SP1 355](#_Toc345624525)

Authoring for System Center 2012 - Operations Manager

The Authoring Guide for System Center 2012 – Operations Manager provides information on creating custom monitoring for your application.

Prerequisites

This guide assumes that you have a basic knowledge of Operations Manager and understand all of the concepts presented in the [Operations Guide for System Center 2012 - Operations Manager](http://go.microsoft.com/fwlink/p/?LinkID=207751).

Not included in this guide

While the following topics are relates to authoring, they are covered in the [Operations Guide for System Center 2012 - Operations Manager](http://go.microsoft.com/fwlink/p/?LinkID=207751) instead of this guide. This is because these features are available to users from the Monitoring workspace of the Operations console and may be used by a user without Author privileges or custom monitoring requirements.

 Overrides

 Views and Dashboards

Author’s Guide Topics

 [Key Concepts for Authors](#z5bb994ba44f54a8386ac9db9202b67ba)

Describes basic concepts related to authoring. You should understand the concepts in this section before reviewing the other content in the guide.

 [Targets and Objects](#z95ee4b5c66394fbf93a2e70057fc8f59)

Describes basic concepts and procedures related to targets and objects.

 [Management Pack Templates](#z7f7e6aa7678e4436841b0e097ef4e4ca)

Describes each of the standard management pack templates which provide complete monitoring scenarios with minimal effort.

 [Monitors and Rules](#zdc82139bfb7f47b5ae3fcafeecf4d4ff)

Describes the structure of monitors and rules and provides details for creating different kinds monitoring scenarios.

 [Distributed Applications](#z9deed7cf5546485f9276a1e4ad9dd928)

Describes how distributed applications work and how to create one using the Distributed Applications Designer.

Downloadable Documentation

You can download a [copy of this technical documentation from the Microsoft Download Center](http://go.microsoft.com/fwlink/?LinkId=246682). Always use the TechNet library for the most up-to-date information.

Key Concepts for Authors

This section of the Author’s Guide to Operations Manager provides general concepts that an author should understand. The subsequent sections of the guide assume familiarity with these core concepts.

Key Concepts Topics

 [What is an Author?](#zb1d1315456614290a3de8bddee57f25f)

Describes the concept of authoring and the tools and permissions required for the author.

 [Authoring Tools](#zadb418d795ab4e338ced34a934016aa3)

Provides an overview of the different tools that are available to the author with an explanation of who should use which tools under different conditions.

 [Structure of a Management Pack](#z801a96149c50414db66f700a18c1aa85)

Describes the basic structure of a management pack and the different kinds of management packs that are available.

 [Sealed Management Pack Files](#zca985c60b85441ff8d4019891d07d6ef)

Describes the implications of sealing a management pack and the procedure for sealing one.

 [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f)

Provides an overview of the concepts of classes and objects.

What is an Author?

A variety of management packs are available that include complete monitoring for particular applications. You can monitor these applications by installing and configuring the appropriate management pack.

Operators can customize certain settings in existing monitoring configurations by using overrides. Overrides can perform such actions as disabling a particular rule, controlling how often a rule runs, or changing the threshold on a monitor. However, overrides cannot be used to create any additional monitoring.

There are times where you might have to add additional monitoring scenarios to an existing management pack or create monitoring for an application or device that has no management pack. When you create new monitoring for an application, you are acting as a Management Pack Author. As an author, you can create additional monitoring scenarios for an application with an existing management pack or create an entirely new management pack for an application that does not have a management pack.

What tools do I use?

Because management packs are implemented in .xml files, any XML editor can create and modify the XML code, however, this is the most complex method. Generally, you can create any monitoring that you require by using much simpler methods in the Operations console. When using the console, you have to select which management pack file you want to use to store any elements that you create, but there are few other details about the management pack that you have to consider.

For more information about the tools that you can use to create a management pack, see [Authoring Tools](#zadb418d795ab4e338ced34a934016aa3).

What permissions do I require?

The permissions that you require to perform authoring depend on the method that you are using. Permissions in Operations Manager are controlled through user roles. All authoring performed in the Operations console requires access to the Authoring workspace. To access this workspace, you must have the Author or Administrator user role. Your authoring credentials might be limited to particular target classes depending on the author scope of the user role. If this is the case, you only can author elements against these classes. For more information about user roles, see [Implementing User Roles](http://go.microsoft.com/fwlink/?LinkID=232869).

If you are using one of the offline methods for authoring, the Authoring console or an XML editor, you do not require any permissions because you are simply creating a file offline. In contrast, to install the management pack, you must have the Administrator user role.

See Also

[Authoring Tools](#zadb418d795ab4e338ced34a934016aa3)

[Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f)

[Implementing User Roles](http://go.microsoft.com/fwlink/?LinkID=232869)

Authoring Tools

There are multiple tools available for creating new monitoring in System Center 2012 – Operations Manager. The choice you make will depend on your requirements and experience with Operations Manager. Each of the available tools is described below.

Operations Console

The Authoring workspace of the Operations console contains templates and wizards that let you create monitoring scenarios that require minimal knowledge of authoring concepts.

Specific characteristics of the Operations console include the following:

 Creates or changes a management pack file in an existing management group. Changes are committed and deployed to agents immediately.

 Creates predefined monitoring scenarios by using templates and wizards. Cannot create custom monitoring scenarios.

 Cannot create custom classes or discoveries except for specific cases that the template provides.

The primary advantage of the Operations console is the ability to create new monitoring scenarios with minimal complexity. The Operations console provides the following approaches for creating monitoring scenarios.

Management Pack Templates

Management Pack templates let you create complete monitoring scenarios with minimal input. A single template can create different monitors, rules, and even new target classes without requiring you to know any of the details. If a template is available for the monitoring scenario that you want to create, this method is typically your easiest and most effective solution.

Distributed Application Designer

The Distributed Application Designer lets you create a single application that is comprised of multiple objects. It does not let you create new monitoring for these objects. The health of each object included in the distributed application is based on the monitors already running against that object. The health of the distributed application is based on the health of the individual objects that are included in it.

Authoring Wizards

Wizards let you create new monitors, rules, and tasks. You have to know the details of what data the workflow uses and what action are performed with that data. Wizards are available for a variety of different data sources and monitoring scenarios. You cannot create a new target with the monitoring wizards. You can use a suitable wizard that is available in an existing management pack, or you can create a wizard by using a template or by editing the management pack in the Authoring console.

Use of the Operations console is documented in this guide.

Visio Management Pack Designer

The Visio Management Pack Designer (VMPD) is an add-in for Microsoft Visio 2010 that allows you to create a management pack using graphical shapes. You can create a custom model representing your application based on a set of common patterns. Standard monitoring scenarios can be added to the model by simply setting properties on shapes that are added to the diagram. You can use the VMPD to create a complete management pack for an application with basic monitoring requirements or use it as a starting point for a management pack that can be edited using other tools.

Specific characteristics of the Visio Management Pack Designer include the following:

 Graphical tool that allows you to create sophisticated management packs while requiring minimal knowledge of the models and monitoring being created.

 Create new classes representing your application that cannot be created in the Operations console.

 Creates predefined monitoring scenarios. Cannot create custom monitoring scenarios.

 Generate a management pack file that can be installed in a management group or used as a starting point for additional modification using other authoring tools.

 Cannot import the management pack back into Visio once it has been generated.

 Creates or changes a management pack on disk without requiring access to the management group.

Documentation for the Visio Management Pack Designer is available at [Visio Management Pack Designer for System Center 2012 - Operations Manager](http://go.microsoft.com/fwlink/?LinkID=232308).

System Center Operations Manager 2007 R2 Authoring Console

The System Center Operations Manager 2007 R2 Authoring console can be used to create management packs for both System Center Operations Manager 2007 R2 and System Center 2012 – Operations Manager. It is intended for management pack authors with significant knowledge of the structure of management pack elements and lets them create and modify all elements in a management pack. It is not limited to a specific set of scenarios, although it does require deeper technical knowledge than the Operations console.

The Authoring console provides many of the same wizards as the Operations console, except that the Authoring console wizards can provide access to additional options. The Authoring console provides custom dialog boxes similar to the Operations console to help you configure management pack elements where they are available. For those elements where no dialog box is available, you must edit the XML code of the management pack directly. To help edit, the Authoring console starts an external editor. You use the editor to edit the XML code of the specific element and, when it is closed, the editor returns the completed element back to the management pack. You can specify the editor that the Authoring console starts.

Specific characteristics of the Operations console include the following:

 Creates or changes a management pack on disk without requiring access to the management group. Optionally, loads management packs from a management group and installs them after modification.

 Provides custom ID for each management pack element.

 Creates all management pack elements including custom classes, relationships, and discoveries that cannot be created in the Operations console.

 Creates predefined monitoring scenarios by using wizards. Creates custom monitoring scenarios with the assistance of the user interface.

Use of the System Center Operations Manager 2007 R2 Authoring Console is documented in the [System Center Operations Manager 2007 R2 Authoring Guide](http://go.microsoft.com/fwlink/?LinkID=188119).

Visual Studio Authoring Extensions

Because management packs are .xml files, any XML editor can create and modify them. While this is more complex than using the other authoring options, editing the XML gives you complete control over all elements in a management pack.

An XML editor is required for the following scenarios that cannot be performed in the consoles:

 Management pack elements cannot be copied by using the other consoles. For example, you might want to create a management pack element by copying a similar element in the same management pack, or you might want to copy an element from another management pack into the current one. This functionality can only be performed with an XML editor.

 The consoles do not let an existing management pack element be modified if it affects the configuration of another element. For example, a rule might use a custom module that requires values for one or more parameters. You might want to modify this module and add an additional parameter to it. Because the rule would not provide a value for this required parameter after the module was modified, the changed module cannot be saved. To make this change in the Authoring console, you must delete the rule and create it again after changing the module. By using an XML editor, you can change the module and rule at the same time.

 The ID of a management pack element cannot be changed with the Authoring console after it has been created. You must perform this task with an XML editor. The ID of the element cannot only be changed, but you must search for the old ID and replace it with the new ID across the management pack because the element’s ID might be used multiple times in the XML code.

The Visual Studio Authoring Extensions allow you to work with the XML of a management pack in System Center 2012 – Operations Manager directly using Microsoft Visual Studio. It provides the following advantages over using a standard XML editor:

 Provides XML templates and IntelliSense for different management pack elements so that you don’t have to have detailed knowledge of the schema.

 Allows you to create XML fragments containing different management pack elements. The fragments can be copied within the management pack, to another management pack, and combined to build the final management pack.

 Allows multiple authors to work on a single management pack project at the same time.

Documentation for the Visual Studio Extensions is available at [Visual Studio Authoring Extensions for System Center 2012 - Operations Manager](http://go.microsoft.com/fwlink/?LinkID=232334).

See Also

[System Center Operations Manager 2007 R2 Authoring Resource Kit](http://go.microsoft.com/fwlink/?LinkID=192098)

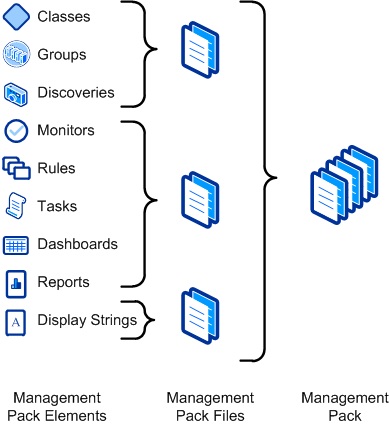
Structure of a Management Pack

Management packs to Operations Manager are the equivalent of games to a video game console. The main function of the console is to provide the basic functionality required to run the game, but the game provides the true value of the overall system. A variety of games are available that each provides a distinct entertainment experience. Similarly, Operations Manager provides basic functionality such as deployment of agents to managed computers and central collection, and analysis of collected information. A management pack is then required for each application that you want to monitor. The management pack only has to provide logic specific to the application and can take advantage of the platform provided by Operations Manager.

Management Pack Files

A single management pack is comprised of one or more management pack files. A management pack file is either an .xml file that can be viewed and edited, or an .mp file which is a sealed .xml file that cannot be changed. When you download a management pack from the management pack catalog, it typically includes a set of management pack files that each contains a specific set of elements. When you create a new element in the Operations console, you must select a management pack file to store it in.

Conceptual view of management pack files and elements



Management Pack Elements

A management pack element is a single item in a management pack such as a class, monitor, dashboard, or report. There are reasons that management pack authors group different elements in a particular file, but you can ignore the reasons until you get to more advanced concepts. For now, just understand that the word element refers to one of these items.

Compatibility with Operations Manager 2007

Operations Manager supports management packs that use a later version of the XML schema than Operations Manager 2007. Management packs written for Operations Manager 2007 work without changes in Operations Manager. However, management packs written by using new Operations Manager elements or using the updated schema do not work in Operations Manager 2007. This includes management packs that are created in the Operations Manager 2012 Operations console because they are based on the new schema.

Types of Management Packs

All management packs use the same schema and can contain any elements that are used in Operations Manager. Management packs with a specific set of elements are referred to with a more specific term as described below.

Monitoring Pack

Monitoring packs are management packs intended to perform general monitoring scenarios. These are the primary management packs obtained from the management pack catalog. They also are the primary management pack that you create with the Operations console and Authoring console.

Feature Pack

Feature packs contain additional features for a monitoring pack. These can be features that are optional. You can select which set of features you want to install. Another reason for a feature pack is that it can contain elements specific to Operations Manager so that the monitoring pack can be used in Operations Manager 2007 and Operations Manager with the additional features only used in Operations Manager.

Dashboard Packs

Dashboard packs are feature packs that contain dashboards which are a new feature to Operations Manager, so they cannot be imported into an Operations Manager 2007 management group. Dashboards are not included in monitoring packs so that the monitoring pack can be used with both version of Operations Manager.

Security Packs

Security Packs are feature packs that monitor the security of an application.

Language Packs

Any text displayed by a management pack in the Operations Console is defined in a display string. Wizards populate the display strings, so you do not have to distinguish them from the elements that they are associated with. Some management pack files contain only display strings in a particular language. These are referred to as language packs. If you require a management pack in alternate languages, you can download the appropriate language packs for the languages that you require. If you are creating your own management pack, you can use the Authoring console or XML code to define display strings in any languages that you require.

Management Pack Schema

Knowledge of the schema for a management pack is only required if you edit the management pack directly by using an XML editor. You are never exposed to this schema when using the Operations console and only rarely with the Authoring console. Wizards are available for creating common elements, and custom user interface pages are available for common modules. Direct XML editing is only required for configuring modules and workflows that do not have custom UI pages. The XML for defining complex data types might be required, and the schema for these data types is provided in the [Schema Type Reference](http://go.microsoft.com/fwlink/?LinkID=192053).

See Also

[Sealed Management Pack Files](#zca985c60b85441ff8d4019891d07d6ef)

Sealed Management Pack Files

When a management pack file is sealed, it is converted to a binary file that has an .mp extension. Either the sealed or the unsealed version of a management pack file can be installed to a management group, but both cannot be installed at the same time.

Important

Sealing a management pack is not a valid strategy for hiding information from users. You can export any sealed management pack from the management group by giving you full access to its XML code. A management pack should never contain sensitive information such as passwords.

Characteristics of Sealed Management Pack Files

Sealed management pack files have the following characteristics:

The contents of the management pack file cannot be modified

Sealed management pack files cannot be changed. Changes must be made to the .xml file which is then sealed again with the same certificate. Such an update can only be installed in the same management group if the updated management pack is backward compatible.

Enforce version control

Only sealed management pack files enforce version control when an updated version of the management pack is installed. If the management pack is unsealed, the new version is always installed regardless of its backward compatibility.

Enable the management pack to be referenced by other management packs

Management packs can only reference an element in another management pack if the management pack that is referenced is sealed. This requirement ensures that a modification to a management pack cannot break other management packs that reference it. Because sealed management packs maintain version control, any referencing management packs ensures that updates to the sealed management pack are backward compatible.

When to Seal a Management Pack File

Management pack files do not all have to be sealed. You can install the unsealed version of a management pack file in a management group, and it behaves exactly as the sealed version of the same management pack file. The following criteria list when a management pack file must be sealed:

 Management pack files that are referenced by other management pack files must be sealed. You might want to create common elements such as groups or modules that are used by other management packs for different applications. The application management packs do not have to be sealed, but the management pack files that contain the shared elements must be.

 Any management packs that are sent to external customers must be sealed. In addition to ensuring that the customer cannot modify the contents of the management pack, it ensures that any modifications they make are made through overrides in a different management pack file. This lets you provide updates to the management pack without affecting the customer’s modifications.

 Management pack files that are shared by multiple business units in your organization should be sealed. This ensures that each business unit makes any modifications through overrides in their own management pack files. Each business unit cannot make modifications that affect the other group.

Sealing a Management Pack File

Management pack files are sealed by using the MPSeal tool that is located in the SupportTools folder of the Operations Manager 2007 R2 distribution media. Sealing requires a client certificate to validate the identity of the author.

You can use the Strong Name tool (Sn.exe) included with the Microsoft .NET Framework SDK to create a certificate sufficient for sealing management packs for testing. For production, use a client certificate from the correct certification authority (CA) appropriate for signing code for sealing management packs.

For information about the process for sealing a management pack file, see [How to Seal a Management Pack File](#z726bf2172e6646218d4e988f1fb4b67d).

See Also

[How to Seal a Management Pack File](#z726bf2172e6646218d4e988f1fb4b67d)

[Structure of a Management Pack](#z801a96149c50414db66f700a18c1aa85)

How to Seal a Management Pack File

Management packs are sealed by using the MPSeal tool that is located in the SupportTools folder of the Operations Manager 2007 R2 distribution media. This is a command-line tool that creates a sealed .mp file from an unsealed .xml file. After performing the sealing process, you can install the sealed management pack in your management group.

Note

If you created the management pack in the Operations console, you must export it to an .xml file before performing the sealing process. You must then uninstall the management pack before installing the sealed version.

MPSeal Syntax

MPSeal.exe uses the following syntax:

MPseal.exe Management Pack File Name [/I Include Path]\* /Keyfile Key File Path /Company Company Name [/Outdir Output Directory] [/DelaySign] [/Copyright Copyright text]

Each of the command-line options are described in the following table.

|  |  |
| --- | --- |
| Option | Description |
| Management Pack File Name | The full name of the .xml file to seal. If the file is not in the current directory, you must include the full path to the file. If the path includes a space, you must enclose it in quotes. |
| Include Path | The path to a directory containing .mp files that are referenced by the management pack that you are sealing. For more information, see [Management Pack References](#z6). |
| Key File Path | The file containing the private and public key. For more information, see [Key File](#z7). |
| Company Name | The name of your company. If it includes a space, you must enclose it in quotes. |
| Output Directory | The directory to store the output file. If not specified, the current directory is used. |
| DelaySign | If this option is used, only the public key is used. For more information, see [Delayed Signing](#z8). |
| Copyright text | Text to include for copyright information. While this option is functional, the text is not currently accessible from Operations Manager. |

Example

The following example seals a management pack file named Contoso.MyApp.xml. It creates a file called Contoso.MyApp.mp in the current directory.

mpseal Contoso.MyApp.xml /I c:\mp /Keyfile contoso.snk /Company "Contoso"

Management Pack References

In addition to sealing the management pack, MPSeal verifies the management pack file and reports any errors that prevents it from installing. All of these errors must be corrected before the sealing finishes successfully. The MPVerify tool performs the same verification . To perform this function, MPSeal requires access to any management packs referenced by the management pack that is in the process of being sealed. These must be the sealed versions of the files with an .mp extension and must be at least the version specified by the management pack that is being sealed.

You specify a directory to search .mp files with the /I command-line option. You can use multiple /I options if the required files are in multiple directories. You can obtain the standard library management pack files included with Operations Manager 2007 R2 from the installation directory on the management server. You must obtain other management pack files separately. If you import a management pack directly into your management group from the management pack catalog, you have to download it separately to obtain the .mp file.

Note

If you are unsure of the management packs referenced by the management pack that you are sealing, you can run MPSeal by using any directory. A list of the required management packs are returned.

For more information about management pack references, see the Management Pack References section of this guide.

Key File

Sealing requires a key file that contains a private and public key. The key pair validates the identity of the signing party and ensures that a malicious user cannot provide a sealed management pack by impersonating someone else. This is the same key pair used for signing .NET assemblies and can be created with the Strong Name tool (sn.exe) included with the [Microsoft Windows SDK](http://go.microsoft.com/fwlink/?LinkID=231265).

Important

You should protect any key file that is used to seal a management pack. If someone else were to obtain this key file, they could seal a management pack by impersonating the original author.

For information about the complete use of the Strong Name tool, see [Sn.exe (String Name Tool)](http://go.microsoft.com/fwlink/?LinkID=231266). The following example is sufficient for most management packs and creates a key file called contoso.snk in the local directory.

sn –k contoso.snk

Delayed Signing

For added security of their private key, organizations often implement a delayed process for signing assemblies. This allows access to the private key to only a few individuals. Using this process, you sign the assembly with only the public key, and then complete the signing with the private key just before a release.

If your organization has an existing process for performing delayed signing of assemblies, you should use this process to seal your management pack for production. You can perform the initial partial sealing of the management pack by using the /DelaySign option.

For more information about delayed signing of assemblies, see [Delay Signing an Assembly](http://go.microsoft.com/fwlink/?LinkID=231267).

Management Pack Files Created in the Operations Console

Management pack files created in the Operations console are unsealed. You can use the following procedures if you have to seal a management pack that you created in the Operations console.

To seal a management pack file that was created in the Operations console

|  |
| --- |
| 1. Export the management pack file to an .xml file. For more information, see How to Export an Operations Manager Management Pack.  2. Seal the XML code. For more information, see [MPSeal Syntax](#z5).  3. Delete the management pack file from the management group. For more information, see How to Remove an Operations Manager Management Pack.  4. Import the .mp file created by MPSeal. For more information, see How to Import an Operations Manager Management Pack. |

See Also

[Sealed Management Pack Files](#zca985c60b85441ff8d4019891d07d6ef)

Selecting a Management Pack File

When you create any monitoring in the Operations console, you have to specify a management pack file for the elements that you are creating. This topic describes a basic strategy that you can follow and provides additional details to help you understand the logic of the recommended strategy.

General Strategy

For applications that already have a sealed management pack installed, typically management packs installed from the Management Pack Catalog:

 Create a separate management pack file to store overrides and new monitoring for that application.

For applications that do not have a sealed management pack installed, typically management packs that you created yourself:

 Create a separate management pack file for each application. Use this file to store overrides and any new monitoring for that application.

For common elements that are used by other management pack files, such as groups:

 Create a separate management pack file for each logical set of elements. Seal this management pack file before installing it.

Note

If you created the management pack file in the Operations console, you must export it to an .xml file, and then seal it. You must then uninstall the unsealed management pack file from the management group before you install the sealed management pack file.

Default Management Pack

The Default Management Pack file contains common elements such as views at the top level of the Monitoring workspace. This is an unsealed management pack file so that you can create views and folders at this level. It should not be used for any other purpose. For creating elements such as monitors and rules, create a new management pack file.

Logically Grouping Elements

Although you could simply create a single management pack file to store all custom elements that you create, it is not a best practice. While management pack elements are treated individually by the agents that run them, the management group works with the management pack file. When a management pack file is installed in the management group or removed from it, it includes all of its management pack elements.

When you determine how to group different elements, take the following considerations into account:

 Management pack files are delivered to any agent computer that requires at least one element in the file. If you use a single management pack file for different applications, elements might be delivered to agents that do not require them. The agent only actually loads the elements for the applications that it has installed, but the entire management pack file is delivered. Breaking up management pack files according to the elements that are relevant to a single application ensures the most efficient delivery of the files to agents.

 You can remove an application from your environment and no longer require its management pack. Or you can obtain a new management pack for an application and want to remove custom monitoring that you implemented. In cases like these, you can uninstall all of the elements for a particular application by removing any of its management pack files. If you combine elements for multiple applications, you limit your ability to manage the monitoring logic for a single application.

 You can develop and test some monitoring logic in a lab environment before moving it into a production management group. Combining elements for a particular application into a single management pack lets you manage that file through the different environments without affecting the monitoring for other applications.

By following the recommend strategy for logically grouping management pack elements, you can ensure that your management group runs as efficiently as possible and can most effectively handle future changes.

Sealed and Unsealed Management Pack Files

When selecting a management pack file, you must consider the implications of sealed and unsealed management packs. An element in one management pack file cannot refer to an element in another file if the file being referenced is not sealed. For this reason, you might have to group-related elements in a single management pack file or seal management pack files meant for general use. For more information about the effects of sealing a management pack, see [Sealed Management Pack Files](#zca985c60b85441ff8d4019891d07d6ef).

Because a sealed management pack file cannot be modified, you can only store new management pack elements in unsealed files. Any management pack created in the Operations console is unsealed, and any dialog box prompting you for a management pack only includes unsealed files.

For example, you might create a set of groups that represent different aspects of your computing environment such as the data center that certain computers reside in, the support personnel that manage particular computers, or the applications that different computers support. You want to use those groups to override monitors and rules that you created in different management pack files.

If you used the Operations console to create the groups in this example in an unsealed management pack file, you could not use them with other management pack files. You have to use one of the following two strategies to implement this solution:

 Create groups in each management pack file with the overrides. This has the advantage of being easy to implement without any requirement to seal a management pack file, but it has the disadvantage of requiring you to potentially create multiple copies of the same group.

 Create a separate management pack file for the groups. After you create the groups in the Operations console, export the management pack to an .xml file, and then seal the .xml file by using the process described in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). You can then install the sealed version of the management pack file so that the groups are available to any other management pack.

See Also

[Management Pack Templates](#z7f7e6aa7678e4436841b0e097ef4e4ca)

[Monitors and Rules](#zdc82139bfb7f47b5ae3fcafeecf4d4ff)

Targets and Objects

An object is the basic unit of management in Operations Manager. An object typically represents something in your computing environment, such as a computer, a logical disk, or a database. A class represents a kind of object, and every object in Operations Manager is considered an instance of a particular class. A target in the Operations console represents all instances of a particular class.

The following sections provide further detail on these concepts and guidance for selecting target and created new classes.

Targets and Objects Topics

 [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f)

Basic concepts for understand classes and objects that are required for basic and advanced authoring.

 [Selecting a target](#z0628456991784ccfb068df0b9b12eee9)

Guidance for selecting a target when you create a monitor

 [Creating a new target](#z22b0f17b5dba4ccf9a6c307b161b5570)

Strategies for creating a simple class to use as a target for custom monitors and rules.

 [Attributes](#zff46b3ca8afc4470ab84cd741f3791c5)

Concepts of how attributes are used and procedure for adding new attributes to an existing class.

Understanding Classes and Objects

To start authoring in Operations Manager, you should understand some basic concepts and terminology about how monitored objects are implemented. If you have used the Operations Manager for monitoring, you are already familiar with using these objects, but for authoring, you should understand them at a deeper level.

Object

An object is the basic unit of management in Operations Manager. An object typically represents something in your computing environment, such as a computer, a logical disk, or a database. It could also represent something more abstract, such as an application, an Active Directory domain, or a DNS zone. An object can also be referred to as an instance of a particular [Class](#z11).

Objects must be discovered before they appear in the Operations console. Management packs include discoveries that inspect information about the agent computer to determine whether an object should be created and what the values for its properties should be. These discoveries are typically run on a schedule, and different management packs define different schedules for different objects. One discovery might run one time per day whereas another might run every few hours.

For more information about discoveries for advanced authors, see [Discovery](http://go.microsoft.com/fwlink/?LinkID=232861) in the [System Center Operations Manager 2007 R2 Authoring Guide](http://go.microsoft.com/fwlink/?LinkID=188119).

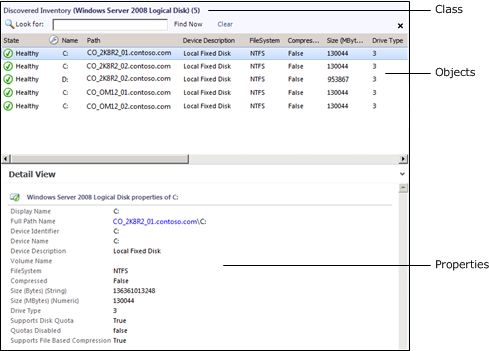
Class

A class represents a kind of object, and every object in Operations Manager is considered an instance of a particular class. All instances of a class share a common set of properties. Each object has its own values for these properties which are determined when the object is discovered. Most management packs define a set of classes that describe the different components that make up the application that is being monitored and the relationships between those classes.

A target in the Operations console represents all instances of a particular class. For example, a view lists all of the objects that are instances of the class that is used as the target class for the view, and a monitor is applied to all objects that are instances of the monitor’s target class.

The following screenshot shows an example of Windows Server 2008 Logical Disk class and objects in the Discovered Inventory view in the Operations console.

Sample Class and Objects



Base Classes

You only have to understand the concept of base classes if you are performing advanced authoring. For more information, see Classes and Relationships. For basic authoring, you only have to understand how base classes affect targeting.

Every class in Operations Manager has a base class. A class has all the properties of its base class and potentially adds more. All of the classes from the different management packs installed in your management group can be arranged in a tree with each class positioned under its base class. If you start at any class, and then walk up the tree following its base class, and then the base class of that class, and so on, you eventually reach the Object class which is the root of the System Center class library.

Note

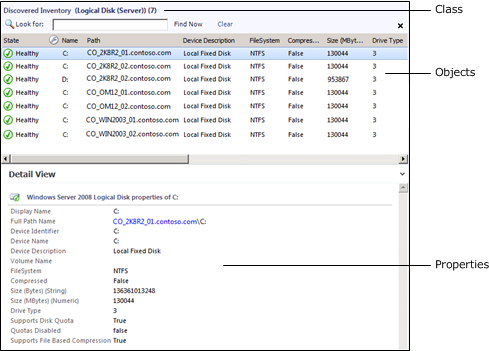
The Entity class is the root of the class library in Operations Manager 2007. The Entity class is the same class as Object. It has been renamed in Operations Manager 2012.

When you select a class as a target that is a base class for other classes, the monitor or rule applies to all instances of each of those classes. For example, if you use Windows Operating System as the target for a monitor, then the monitor applies to all instances of Windows Client Operating System and Windows Server Operating System. This is because those two classes use Windows Operating System as their base class. If you use Operating System as the target, the monitor applies to all those classes and also to any instance of Unix Operating System.

For information about how to view the complete class library in your management group from the Distributed Application Designer, see [Distributed Applications](#z9deed7cf5546485f9276a1e4ad9dd928).

The following screenshot shows an example of Logical Disk (Server) class and objects in the Discovered Inventory view in the Operations console. This is the base class for Windows Server 2008 Logical Disk which is shown earlier in this section. Note that this includes the same set of objects, but adds instances of Windows Server 2003 Logical Disk which also uses Logical Disk (Server) as its base class.

Sample Base Class and Objects



Hosting Classes

Most classes are hosted by another class. When one class hosts another, the hosting class is called the parent, and the class being hosted is called the child. Instances of the child class cannot exist without a parent. For example, several classes are hosted by Windows Computer because they are components on a computer. It would not make sense to have a logical disk if there was no computer for the disk to be installed on. Therefore, Logical Disk is hosted by Windows Computer. This means that every instance of Logical Disk must have one instance of Windows Computer as its parent.

Hosting classes can be important because the properties of the host of a target class are available in rules and monitors. An object might also have the health of its child objects available in its Health Explorer.

You can typically determine the hosting parent of an object by viewing its Path Name property. This includes the name of the object’s parent. In the Operations console, you can click this name to open a state view for the parent object. In the previous screenshots, the name of the computer hosting the selected logical disk is included in the selected object’s path name.

Group

A group is a collection of objects. They can be instances of the same class or of different classes. Groups have population criteria that define what objects are added to them. This can be dynamic criteria that adds objects as they are discovered or explicit criteria where you manually add specific objects to the group.

Groups are used to scope overrides, views, and user roles and to set the scope of monitoring for certain templates. They are not used to target monitors and rules. For more information, see [Targeting a group](#z15),

Viewing Classes and Objects

Use the following procedure to walk through the examples illustrated earlier in this section in your own Operations Manager environment. This shows the concepts of objects, classes, and base classes by using the Logical Disk classes in the Windows Server Operating System Management Pack. You must have this management pack installed in your management group to complete this procedure.

To view a class in the Operations console

|  |
| --- |
| 1. Start the Operations console.  2. In the navigation pane, click Monitoring, and then select Discovered Inventory.  3. In the Actions pane, click Change Target Type. In the Select Items to Target dialog box, select View all targets.  This list consists of all the classes included in all the management packs that are currently installed in the management group. Any of these classes can be selected to view a list of all its discovered instances and their properties. Any new classes included in a management pack that is installed later in the management group will be included in this list.  4. Select Windows Server 2008 Logical Disk, and then click OK.  This view shows a listing of logical disks on Windows Server 2008 computers that were discovered in the current environment.  5. Select one of the instances.  Take note of the properties in the Detail View pane. This shows the values for each property that were collected by the discovery process. Notice also the Path name property that is built from the key property of the current class and its parents. In this case, the key properties include the computer name and the device name.  6. In the Actions pane, again select Change Target Type.  7. In the Select Items to Target dialog box, select View all targets.  8. Select Logical Disk (Server), and then click OK.  This is the class that is the base class for Windows Server 2008 Logical Disk. The view resembles the previous one, but includes objects from both Windows Server 2003 and Windows Server 2008, assuming both are installed in your environment. The properties are identical to the previous view, because the Windows Server Logical Disk class has the same properties that are inherited by Windows Server 2008 Logical Disk.  9. In the Actions pane, again select Change Target Type. In the Select Items to Target dialog box, select View all targets.  10. Select Logical Disk, and then click OK.  This is the class that is the base class for Logical Disk (Server). The instances are identical to the previous view, but fewer properties are shown. This is because the Logical Disk class has only the properties directly assigned to it, and inherits only its single property from Object. The other properties are not visible because they are associated with a class further down the tree.  11. Select one of the disks and then click the link on the computer name part of the Path Name property shown in the Detail View.  This opens a state view for the hosting object of the disk object that you selected. You can see in the Detail View that this is an instance of Windows Computer. |

See Also

[Selecting a target](#z0628456991784ccfb068df0b9b12eee9)

[Creating a new target](#z22b0f17b5dba4ccf9a6c307b161b5570)

Selecting a target

Every time you create a monitor or rule, you must select a target for it. The target determines where the monitor or rule will run and how its information will be organized in the Operations console. It is important to understand how targets work so that you can select the most appropriate one for the monitor or rule that you are creating.

If you are creating a monitor or rule for an application or device that already has a management pack installed, then you can most likely use a target that already exists in your management group. If you are creating a monitor or rule for a new application then you may need to create a new target specifically for this purpose. Creating a new target is covered in [Creating a new target](#z22b0f17b5dba4ccf9a6c307b161b5570).

You do not select specific objects to target in Operations Manager 2012 but rather you select a particular class of object. The monitor or rule will apply to all instances of the target class. For more information on classes and instances of classes, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f).

Quick Logic

If you do not have a full understanding of the logic behind the selection of a target, you can use the following quick logic. Refer to the rest of this section for a complete explanation of the effects of a target and how to select an appropriate one.

Examples

 If you want the monitor or rule to run on all Windows computers in your management group, then select Microsoft Windows Computer.

 If you want the monitor or rule to run on all Unix computers in your management group, then select Unix Computer.

 If you want the monitor or rule to run on all computers running SQL Server 2008 in your management group, then select SQL 2008 DB Engine.

 If you want the monitor or rule to run on all computers running any version of SQL Server in your management group, then select SQL DB Engine.

 If you want the monitor or rule to run against all logical disks on all Windows servers in your management group, then select Logical Disk (Server).

Effects of a target

The target of a monitor or rule determines the following:

 [Where the monitor or rule will run](#z18)

 [How many copies of the monitor or rule will run on the agent](#z19)

 [What object the data will be associated with](#z20)

 [What properties will be available for the expression and the alert description](#z21)

Where the monitor or rule will run

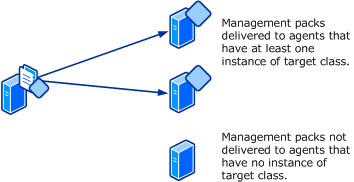
The monitor or rule will run on each agent that has at least one instance of the target class. You should select a target that only includes those agents where you want the rule or monitor to run.

When a management pack is installed or changed, it is delivered to any agent that manages at least one instance of a class that is used as a target on at least one included workflow. Only workflows targeted at classes that have discovered instances on the agent are actually loaded.

You must ensure that the data that the monitor or rule is accessing will be available in the agent where it is running. For example, if you are creating a monitor that looks for a particular event, you must ensure that the event will be written to the event log where the monitor is running. If you are creating a rule that collects a performance counter, you must ensure that the performance counter is available on that agent. If you are creating a monitor that runs a script, then the script will run on that agent and any resources that it access must be available on the local computer.

If you select a very broad class such as Windows Computer, then the monitor or rule will probably run on the agents that you need, but it will also probably run on additional agents where it is not needed. If a broad class such as this is the only one that includes the agents that you need, then consider creating a new target as described in [Creating a new target](#z22b0f17b5dba4ccf9a6c307b161b5570).

Management pack delivery

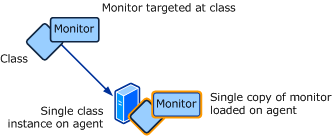


How many copies of the monitor or rule will run on the agent

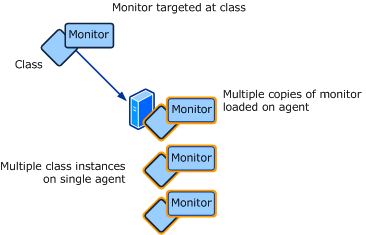
One copy of the monitor or rule will run for each instance of the target. If there is more than one instance of the target on an agent, then more than one copy of the monitor or rule will run on that agent. You can determine how many objects are running on each agent in your environment using the procedure in [Viewing Classes and Objects](#z22).

For targets such as Windows Computer, you can be assured that they will only have a single object on a particular agent. Other targets such as Logical Disk may have only object but could have multiple depending on the configuration of the agent computer.

Workflow targeted at a single instance



Workflow targeted at multiple instances



If you do target a class that may have multiple instances on a single agent, then you need to include criteria in the expression to unique identify each instance.

What object the data will be associated with

Monitors are included in the Health Explorer of their target object. This has implications beyond just the user interface. Any downtime from the monitor would be included in the availability of the target object. For example, while using Windows Computer as a target might cause a monitor to run on the agent computers that you want, any downtime from the monitor would result in downtime for the entire computer, which may not be valid. It would be better to use a target more specific to the component that you want to monitor.

Alerts and data collected by rules will be associated with their target object. When you right-click an open alert and select Health Explorer, you will open the Health Explorer of the target object. When you right -click an object and select Performance View, you will open a view with performance data for the target object. To achieve optimum functionality from the monitors and rules that you create, you should select or create a target that most accurately represents the component that you want to monitor.

What properties will be available for the expression and the alert description

Properties of the target object are available for both any expressions that are used in the monitor or rule and in the description of the resulting alert.

Viewing Classes and Objects

You can use the Discovered Inventory view in the Operations Console to assist in selecting an appropriate target for a particular monitor or rule. This view will list all instances of a specified class with its entire set of properties. This shows which instances of the class are managed by different agents, how many instances each agent is managing, and what properties the class has available. Any workflows targeted at the selected class will be run against each of the objects listed in the view. If no instances are listed for a particular class, then none have been discovered on any agents. Any workflows targeted at these classes would not be run.

You can view the discovered inventory for a particular class using the following procedure.

1. In the Operations console, click the Monitoring button.

2. In the Monitoring pane, select Discovered Inventory.

3. In the Actions pane click Change Target Type.

4. In the Select Items to Target dialog box, select the class to view, and then click OK.

Targeting a group

Groups are included in the dialog box for selecting a target, but you will receive a warning if you select one. The reason for this is that there is rarely a case where you will want to select one. Monitors and rules run on the agent that manages their target object. The workflow will not enumerate the contents of the group but will attempt to run against the group object itself. Since groups are managed by the management server currently running the Root Management Server services, any workflows targeted at them will be loaded only on that server.

If you do want to have a monitor or rule run only on the members of a particular group, then you can use the following procedure. Note that this is not an ideal strategy, and under most circumstances it would be more effective to create a target for your application as described in [Creating a new target](#z22b0f17b5dba4ccf9a6c307b161b5570).

To create a monitor or rule that runs on agents in a group

|  |
| --- |
| 1. Create the monitor or rule and use Windows Computer for the target.  2. Disable the monitor or rule that you just created.  3. Create an override to enable the monitor or rule for the group. |

Selecting a group for a monitoring wizard

Certain monitoring wizards will require a group to be specified. This specifies the group of computers that will be searched to determine if they have the component that the wizard is monitoring. For example, if you run the Windows Service monitoring wizard, you specify the name of a service to monitor. The wizard will search all computers in the target group that have the service installed. Only those computers with the service will be monitored.

Examples

 If you want the wizard to include all Windows agents in your management group, select All Windows Computers.

 If you want the wizard to include all Unix agents in your management group, select Unix Computer Group.

 If you want the wizard to include all computers running SQL Server 2008 in your management group, then select SQL 2008 Computers.

 If you want the wizard to include all computers running any version of SQL Server in your management group, then select SQL Computers.

Creating a new target

There are multiple methods that you can use to create a new class that can be used as a target for monitors and rules in System Center 2012 – Operations Manager. Any classes in a management pack created by the [Visio Management Pack Designer](#z2) can be used as a target. Advanced authors can refer to the [Service Model](http://go.microsoft.com/fwlink/?LinkID=232863) section of the [System Center Operations Manager 2007 R2 Authoring Guide](http://go.microsoft.com/fwlink/?LinkID=188119) for detailed information on creating a complex class model for their application. This advanced information is not required though to create a basic class that can act as a target for monitors and rules specific to a particular application.

Management Pack Templates

The following management pack templates in the Operations console create a class that can be used as a target for monitors and rules:

Windows Service Template

If your application has service a Windows installed on each server, then you should use the [Windows Service Template](#zc9e43d3744604da5b9b95756f1f857ee). This will create a new class and discover an instance on all agent computers with the service installed. If any monitors or rules use this class as a target, then they will run on those same agents.

Process Monitoring Template

If your application does not have a Windows service but does have a process that is running on the agent computer, then you should use the [Process Monitoring Template](#z684a5a24ce6543b8af8e4ade060acf6e). This will create a new class and discover an instance on all computers in a specified group. If any monitors or rules use this class as a target, then they will run on those same agents.

Unix/Linux Service

If your application has a service on a Unix or Linux server, then you should use the [UNIX or Linux Process](#zbca73ff2ae7542d09b12ef37f98e1c7a) template. This will create a new class and discover an instance on all agent computers with the service installed.

Simple class using Authoring Console

The [System Center Operations Manager 2007 R2 Authoring Console](#z3) is typically used by advanced users for custom management packs. It can be used though to create a simple class and discovery that you can then install in your management group and perform further authoring using the Operations console.

In addition to the class, you must create a discovery so that instances of the class can be created on agents where the application is installed. The Authoring Console provides a wizard that creates a discovery based on the Windows registry. This will allow you to specify criteria such as the name of a registry key. If the key is present, then the application is installed, and an instance of the class should be created.

To create a class and discovery in the Authoring Console

|  |
| --- |
| 1. Open the Authoring Console.  2. Select File and then New.  3. On the Management Pack Template page, do the following:  a. In the Select a Management Pack Template pane, select Windows Application (Registry).  b. For the Management Pack Identity, type a name such as Contoso.MyApplication.  Note  This name may not contain spaces and should start with the name of the management pack.  c. Click Next.  4. On the Name and Description page, type a Display Name such as Contoso My Application for the management pack and click Next.  5. On the Windows Application page, do the following:  a. In the ID box, type a unique ID for the new class such as Contoso.MyApplication.MyTarget.  Note  This name may not contain spaces and should start with the name of the management pack.  b. In the Display Name box, type a display name for the new class such as My Application Target  c. Click Next.  6. On the Discovery Schedule page, set the schedule to 1 hour or more and click Next.  Note  This is the frequency that registry criteria on the agent computer will be evaluated to determine if an instance of the class should be created.  7. On the Registry Probe Configuration page, do the following:  Note  On this page, you specify the registry keys and values that your criteria will use.  a. Click Add.  b. Leave the Object Type set to Key.  c. In the Name box, type a name such as KeyExists.  Note  You can use any name that is descriptive. The name is not displayed to the user and is only used on the next page of the wizard.  d. In the Path box, type the path of the registry key to check such as SOFTWARE\MyApplication.  e. In the Attribute Type box, select Check if exists.  f. Click Next.  8. On the Expression Filter page, do the following:  Note  On this page, you specify the criteria for evaluating the registry data collected on the previous page.  a. Click Insert.  b. Click the ellipse button next to Parameter Name and select KeyExists.  c. In the Operator box, select Equals.  d. In the Value box, type True.  e. Click Create. |

To install the management pack from the Authoring Console

|  |
| --- |
| 1. Select Tools and then Export MP to Management Group.  2. Select or type in the name of a management server for your management group.  3. Click Connect. |

See Also

[Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f)

[Selecting a target](#z0628456991784ccfb068df0b9b12eee9)

[Management Pack Templates](#z7f7e6aa7678e4436841b0e097ef4e4ca)

Attributes

Attributes in Operations Manager are properties of a class. Each instance of the class has the same set of attributes, but each instance can have its own value for each attribute. These values are populated when the object is discovered.

Uses for Attributes

The following table lists different ways that attributes can be used.

|  |  |
| --- | --- |
| Management Pack Element | How attributes can be used |
| Views and Dashboards | A view or dashboard can include all objects with a particular value for an attribute. |
| Groups | A group can define criteria for dynamic membership that populates the group with all objects that have a particular value for an attribute. |
| Monitors and Rules | The value of an attribute can be used in the definition of the monitor or rule. For example, a monitor may run a script that requires information from the target object. An attribute could be used as a command line parameter sent to the script. |
| Alert Descriptions | An alert description created by a monitor or rule can include values from the data or from the attributes of the target object. |

Custom Attributes

You may want to add custom attributes to existing classes so that you can collect additional information supporting views and groups in your environment. You can add attributes to any class and populate it with data retrieved from the registry or from a WMI query.

When you create a new attribute, a new class is created based on the existing class. The new class has the new attribute and inherits all of the attributes from the original class, so that it is interchangeable with the original class. In order to use the new attribute, you must select the new class. An instance of the new class will be discovered for each member of the original class and the new attribute populated only on those agents where the specified data is found.

Warning

When you create a new attribute in the Operations console, a new class is created for each custom attribute that you create. Even if you create multiple attributes on the same class, a new class will be created for each one. Too many classes can result in excessive overhead. If you create more than a few custom attributes for a class, you should use another tool such as the System Center Operations Manager 2007 R2 Authoring Console that will allow you to create a single class with multiple attributes.

Wizard Options

When you run the Create Attribute Wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General Properties

The General Properties page includes the name and description of the attribute.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name of the attribute that will be displayed in the Operations console. |
| Description | Optional description of the attribute. |

Discovery Method

The Discovery Method page includes the method for populating the attribute and the class that it will target.

|  |  |
| --- | --- |
| Option | Description |
| Discovery Type | Specifies if the value of the attribute will be populated from the registry or from a WMI query. |
| Target | The class to which to add the attribute. |
| Management Pack | Management pack to store the attribute.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Registry Probe Configuration

The Registry Probe Configuration page includes the details of the registry key or value used to populate the attribute. This page is only displayed if the Discovery Type is Registry.

|  |  |
| --- | --- |
| Option | Description |
| Key Or Value Type | Specifies if a registry key or registry value will be used. The data in a registry value can be used for the attribute, while it can only return a true or false indicating if a registry key exists. |
| Path | The path to the registry key or value. |
| Attribute Type | The type of data stored in the registry value. Check if exists can be selected if you only want to determine if a registry key or registry value exists. |
| Frequency | Specifies how often the discovery of the attribute should run. This should typically be a value of an hour or higher since the values of attributes rarely change. |

WMI Configuration

The WMI Configuration page includes the details of the WMI query used to populate the attribute. This page is only displayed if the Discovery Type is WMI Query.

|  |  |
| --- | --- |
| Option | Description |
| WMI Namespace | The WMI namespace with the class used in the query. |
| Query | The WMI query to run. |
| Property Name | The name of the property from the WMI query with the value to populate the attribute. |
| Frequency | Specifies how often the discovery of the attribute should run. This should typically be a value of an hour or higher since the values of attributes rarely change. |

Creating custom attributes

The following example procedure creates an attribute with the following details:

 New attribute named Location on all instances of Windows Computer.

 Value populated from a registry key at HKLM\SOFTWARE\Contoso\Location.

To create a new attribute using the registry

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then expand Management Pack Objects.  3. Right-click Attributes and select Create a new attribute.  4. On the General page, do the following:  a. In the Name box, type Location.  b. Click Next.  5. On the Discovery Method page, do the following:  a. In the Discovery Type dropdown, select Registry.  b. Click Browse to select the target class.  c. Select Windows Computer and then click OK.  d. In the Management Pack box, select the management pack from step 1.  e. Click Next.  6. On the Registry Probe Configuration page, do the following:  a. For Key or Key Value Type select Value.  b. In the Path box, type SOFTWARE\Contoso\Location.  c. In the Attribute Type dropdown, select String.  d. In the Frequency box, type 3600 seconds.  e. Click Finish. |

The following example procedure creates an attribute with the following details:

 New attribute named Manufacturer on all instances of Windows Computer.

 Value populated from a WMI query using the Win32\_ComputerSystem class.

To create a new attribute using WMI

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then expand Management Pack Objects.  3. Right-click Attributes and select Create a new attribute.  4. On the General page, do the following:  a. In the Name box, type Location.  b. Click Next.  5. On the Discovery Method page, do the following:  a. In the Discovery Type dropdown, select WMI Query.  b. Click Browse to select the target class.  c. Select Windows Computer and then click OK.  d. In the Management Pack box, select the management pack from step 1.  e. Click Next.  6. On the WMI Configuration page, do the following:  a. In the WMI Namespace box, type root\cimv2.  b. In the Query dropdown, type select \* from win32\_computersystem.  c. In the Property Name box, type Manufacturer.  d. In the Frequency box, type 3600 seconds.  e. Click Finish. |

See Also

[Base Classes](#z12)

[Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

Management Pack Templates

Management Pack Templates provide Monitoring wizards that let you create complete monitoring scenarios with minimal input. The wizard creates the required monitors, rules, and even targets to implement the particular scenario. There is no requirement for you to understand the management pack elements that are created. You can modify the configuration of the wizard itself if you want to change the way that monitoring is being performed.

Conceptual view of a monitoring wizards



If a management pack template is available for your particular monitoring requirement, using the template most likely is your best strategy. In many cases, you could create the individual rules and monitors yourself, but this exercise is significantly more complex than using an available template. In addition, some templates perform actions that you cannot perform in any other way in the Operations console.

The following table lists the management pack templates that are part of the standard Operations Manager installation. You can install other management packs that might provide additional templates. Each of these templates is covered in detail in subsequent sections of this guide.

|  |  |
| --- | --- |
| Template | Description |
| [.NET Application Performance Monitoring Template](#z2166e40acc7d4534ad8e73a51ac2bf99) | Monitor .NET applications to get details about application performance and reliability. |
| [OLE DB Data Source Template](#z1602145a623d49da87e07c540486f8b4) | Monitor a database accessible with OLE DB. |
| [Process Monitoring Template](#z684a5a24ce6543b8af8e4ade060acf6e) | Discover and monitor instances of a particular Windows process. |
| [TCP Port Template](#z2a51970361f9426297d8490a7807beef) | Monitor the availability of an application that is listening on a specific port. |
| [UNIX or Linux Log File](#z9859313b910544bc9a50d48814bcb28f) | Monitor a UNIX or Linux log file for a specific entry. |
| [UNIX or Linux Process](#zbca73ff2ae7542d09b12ef37f98e1c7a) | Monitor a UNIX or Linux process. |
| [Web Application Availability Monitoring Template](#z62d3b2b3f4ff4fe180553078179c1b1e) | Monitor the availability of one or more web application URLs and run these monitoring tests from internal locations. |
| [Web Application Transaction Monitoring Template](#z7a62ddcca2d544678b53c595c52f1221) | Monitor the availability, operation, and performance of a web application. |
| [Windows Service Template](#zc9e43d3744604da5b9b95756f1f857ee) | Discover and monitor instances of a particular Windows service. |

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

[Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1)

Creating Management Pack Templates

Use the following procedure to create and modify management pack templates.

To create a management pack template

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, right-click the Management Pack Templates node, and then click Add Monitoring Wizard.  4. Select the management pack template that you want to create, and then click Next.  5. Follow the instructions for the template that you selected. You can use the links in the table above to access this content. |

To edit an existing management pack template

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand the Management Pack Templates node.  4. Right-click the template to edit and then select Properties. |

To view the elements created by the management pack template

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand the Management Pack Templates node.  4. Right-click the template to view, select View Management Pack Objects, and then select the kind of element that you want to view. |

See Also

[Management Pack Templates](#z7f7e6aa7678e4436841b0e097ef4e4ca)

Watcher Nodes

A watcher node is an agent that runs monitors and rules that test an application or feature on another computer. The following management pack templates use watcher nodes:

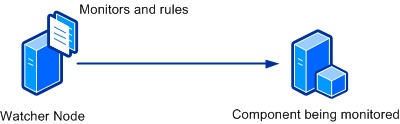
 OLE DB Data Source

 TCP Port

 Web Application Transaction Monitoring

When you use one of these templates, you must specify one or more watcher nodes to run the monitor.

Conceptual view of watcher nodes



A watcher node can either be the agent with the application or feature installed, or it can be a separate agent. If the watcher node is a separate computer, in addition to ensuring that the application or feature is healthy, the watcher node can validate that clients can connect to it and test such additional features as security, network availability, and firewalls.

You can best ensure availability of an application or feature by specifying watcher nodes in different network segments. For example, a database might be accessed from application servers in each of two different network segments. While the database might be available to one set of servers, an issue such a problem with a network device could make it inaccessible to servers on the other segment. If you use the OLE DB Data Source template to create a monitor for the database with a watcher node in each segment, you are assured that any problem accessing the database is detected.

You can also specify the computer with the application or feature itself as a watcher node. This watcher node performs the test without relying on any external features.

See Also

[TCP Port Template](#z2a51970361f9426297d8490a7807beef)

[OLE DB Data Source Template](#z1602145a623d49da87e07c540486f8b4)

[Web Application Transaction Monitoring Template](#z7a62ddcca2d544678b53c595c52f1221)

.NET Application Performance Monitoring Template

The .NET Application Performance Monitoring (APM) template in System Center 2012 – Operations Manager lets you monitor Internet Information Services (IIS)-hosted .NET applications from server- and client-side perspectives to get details about application performance and reliability that can help you pinpoint root causes of incidents. (For System Center 2012 SP1 only: You can also monitor Windows Services.) When you specify settings, the types of events to collect, the performance goals to measure, and servers to monitor, .NET Application Performance Monitoring reveals how applications are running. You can see how frequently a problem is occurring, how a server was performing when a problem occurred, and the chain of events related to the slow request or method that is raising exceptions. This information is required to partner with software developers and database administrators to help ensure that applications perform correctly and reliably for your customers.

This template lets you monitor applications and web services that are hosted in Internet Information Services (IIS) 7.0. You can select one or more applications or services discovered by the IIS 7.0 management pack and configure monitoring of performance and exception events. You must have the Windows Server 2008 Internet Information Services (IIS) 7.0 management pack installed to monitor applications and web services.

For System Center 2012 SP1, you can use the template to monitor applications and web services that are hosted in Internet Information Services (IIS) 8.0. You can select one or more applications or services discovered by the IIS 8.0 management pack and configure monitoring of performance and exception events. You must have the Windows Server 2012 Internet Information Services (IIS) 8.0 management pack installed to monitor applications and web services.

For more information, see [Before You Begin Monitoring .NET Applications](#z43beed71594a40d9aa347210b0e73bcb)

[Scenarios](#z23)

[Monitoring Performed by the .NET Application Performance Monitoring Template](#z24)

[Viewing Monitoring Data](#z25)

[Wizard Options](#z26)

[Server-Side Configuration](#z27)

[Advanced Settings for Server-Side Monitoring](#z28)

[Server-Side Customization](#z29)

[Server-Side Modifying Settings](#z30)

[Transaction Properties: Add ASP.NET Web Page](#z31)

[Transaction Properties: Add ASP.NET Web Service](#z32)

[Transaction Properties: Add Function](#z33)

[Client-Side Configuration](#z34)

[Advanced Settings for Client-Side Monitoring](#z35)

[Enable Client-Side Monitoring](#z36)

[Client-Side Modifying Settings](#z37)

[Summary](#z38)

[Creating and Modifying .NET Application Performance Monitoring Templates](#z39)

[Viewing .NET Application Performance Monitoring Monitors and Collected Data](#z40)

Scenarios

Use the .NET Application Performance Monitoring template in scenarios where you have to monitor web-based applications. These scenarios include the following monitoring processes:

Server-Side Monitoring: Single- or Multi-Tier Web Applications

You might have applications that must be running at all times. Use the .NET Application Performance Monitoring template to ensure that your applications are reliable, have no exceptions, and meet service level agreements (SLAs), in short, that they perform correctly on the computers where they are installed.

Client-Side Monitoring: Browser Performance and Reliability

You want to ensure that your customers are having quality web experiences. By creating or editing existing templates, you can extend your server-side monitoring by adding client-side monitoring that measures the browser experience of your customers.

Monitoring Performed by the .NET Application Performance Monitoring Template

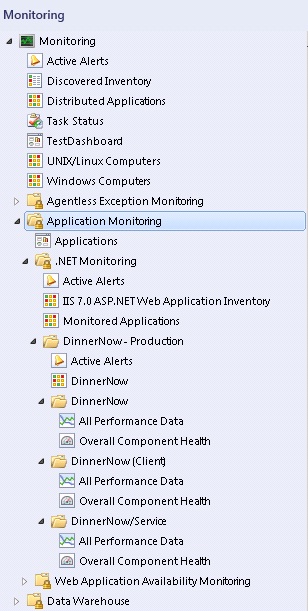
By default, the .NET Application Performance Monitoring template configures the following monitoring. You can be enable, disable, and modify monitors in the Advanced Configuration page of the .NET Application Performance Monitoring template.

|  |  |
| --- | --- |
| Monitor description | Default values |
| Percentage of exception events per monitored requests | Enabled, Threshold=15%, Interval=5 minutes |
| Percentage of performance events per monitored requests | Enabled, Threshold=20%, Interval=5 minutes |
| Average Request Time | Enabled, Threshold=10,000 ms, Interval=5 minutes |

Viewing Monitoring Data

All data collected by the .NET Application Performance Monitoring template appears in the .NET Monitoring folder in the Application Monitoring folder in the Monitoring navigation pane. For each of the application groups that you create by using the .NET Application Performance Monitoring template, the template creates a folder under .NET Monitoring. The Application Monitoring folder contains the default views and subfolders that provide health state, Performance views, and alerts related to the application components in the application group. By using the top-level Application Group State view, you can see the health of the individual components and the monitoring configurations that have been enabled. The state of each object matches the state of the targeted object that has the worst health state so that you see the worst state of the monitors that are running. If one or more of the components are shown with an error while at least one other component is healthy, it could indicate a problem with that particular component, such as a credential issue. If all of the components are unhealthy, it could indicate a problem with the infrastructure, such as network connectivity issues.

Application Monitoring folders

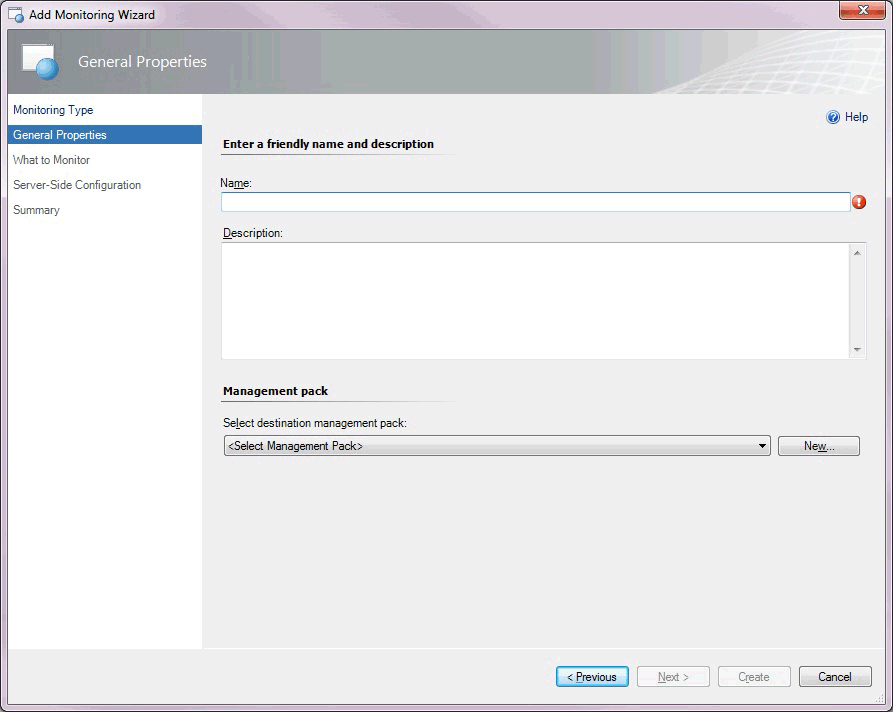


To view the state of the individual monitors, open the Health Explorer for each component. Drill down to the unhealthy monitors to see what is making your application unhealthy. For more information, see [Monitoring .NET Applications](http://go.microsoft.com/fwlink/?LinkID=225673)

Wizard Options

When you run the .NET Application Performance Monitoring template, you have to provide values for options as listed in the following tables. Each table represents a single page in the wizard.

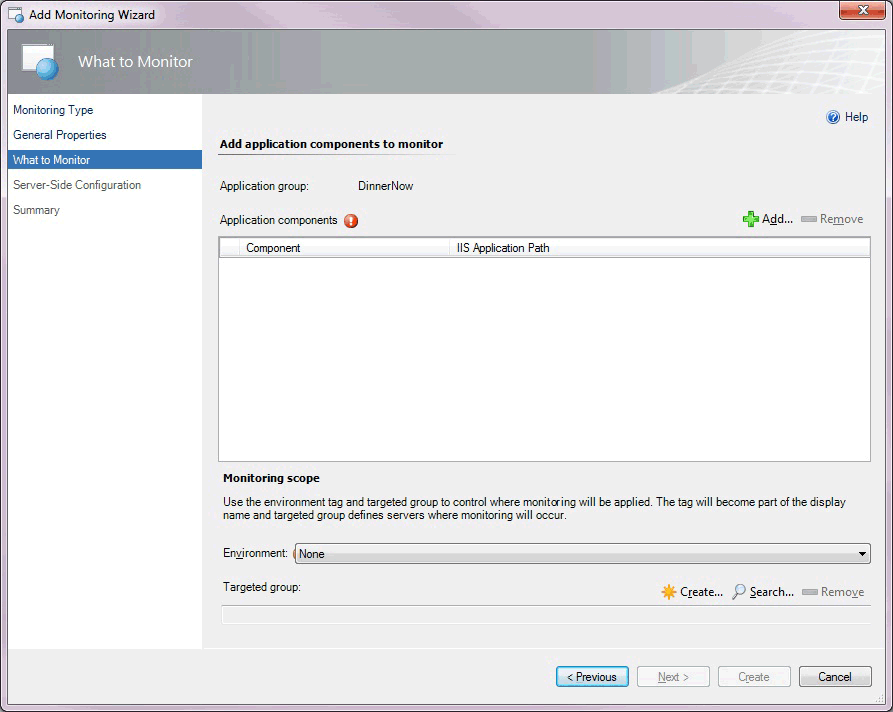
General Properties



The following options are available on the General Properties page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | Enter the friendly name used for the template and application group that you are creating. This name is displayed in the Operations console and used for the folder under the .NET Monitoring folder.  Note  After you have given the template a name and saved the template, this name cannot be edited without deleting and re-creating the template instance. |
| Description | Describe the application group. (Optional) |
| Select destination management pack | Select the management pack to store the views and configuration created by the template. Use the same name for your new management pack as the application group so you can easily pair the two names. You can use an existing management pack or create a new management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

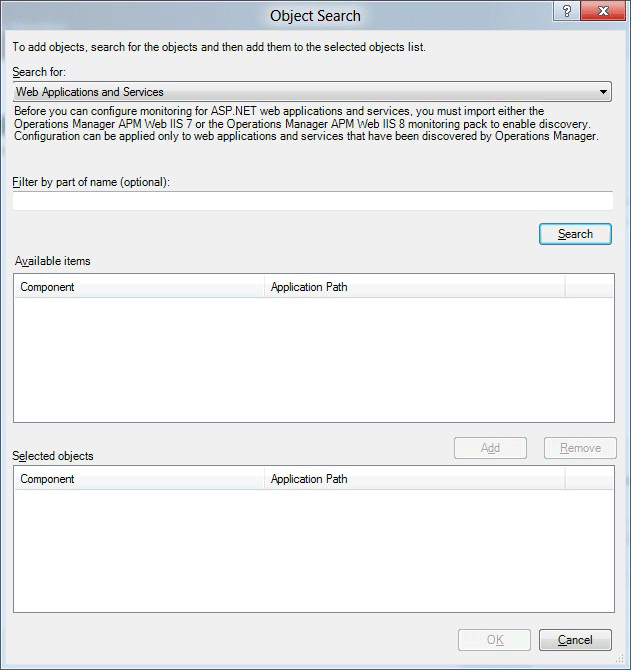
What to Monitor



The following options are available on the What to Monitor page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Application components, Add | Search for and add or remove the application components to monitor.  When you click Add, the Object Search page opens, which lets you select whether you want to monitor Web Applications and Services. For System Center 2012 SP1 only: You can monitor Windows Services.  Note  For System Center 2012 SP1 only: Before you begin monitoring Windows Services, you need to configure Windows Services using the Windows Service template. Once you do this, the.NET Application Performance Monitoring template can discover the Windows Services that are running. For more information, see [Authoring the Windows Service Template](http://go.microsoft.com/fwlink/?LinkId=252385). |
| Environment | Select the environment in which you want to monitor your application: None, Production, Staging, Test, Development, or use New to create a new tag. Typically, you want to pair the environment tag with the server group that you are monitoring. The tag is appended to the application group name and component names, letting you differentiate the event data in Application Diagnostics and Application Advisor. From a monitoring perspective, the environment tag lets you separate the same application into multiple virtual applications.  Note  After you have selected an environment tag and saved the template, the tag cannot be edited without deleting and re-creating the template instance. |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

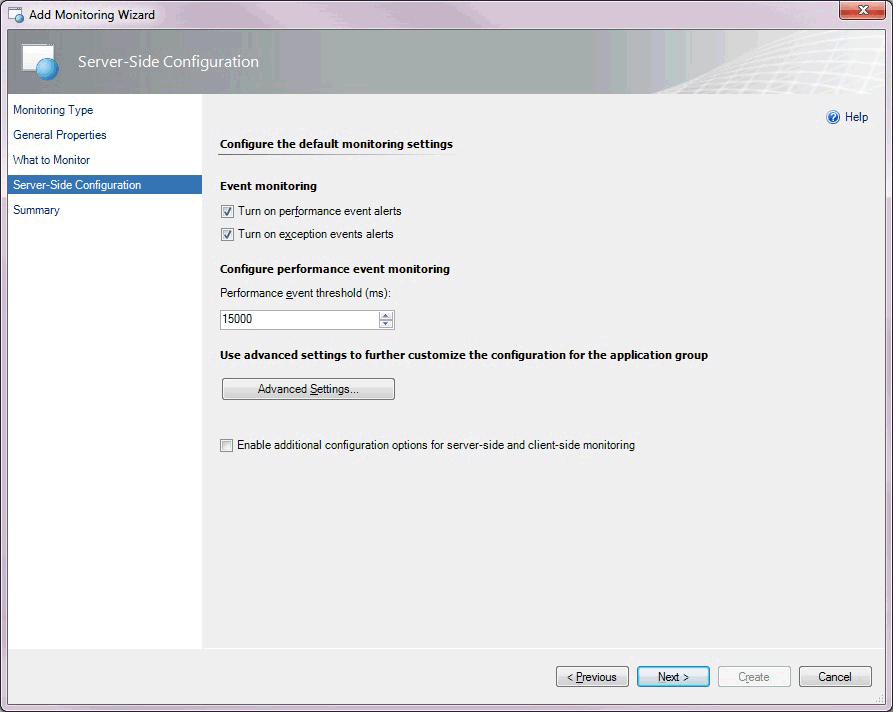
Object Search



The following options are available on the Object Search page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Search for | Select Web Applications and Services. For System Center 2012 SP1 only: You can also select Windows Services.  Note  For System Center 2012 SP1, before you begin monitoring Windows Services, you need to configure Windows Services using the Windows Service template. Once you do this, the.NET Application Performance Monitoring template can discover the Windows Services that are running. For more information, see [Authoring the Windows Service Template](http://go.microsoft.com/fwlink/?LinkId=252385) |
| Filter by part of name (optional) | Enter part of the name of Web Application and Services that you want to select. For System Center 2012 SP1 only: You can also enter part of the name of a Windows Service that you want to select. |
| Available items | Displays the Windows Web Application and Services that are available for monitoring. For System Center 2012 SP1 only: Also displays the Windows Services that are available for monitoring. |
| Selected objects | Displays the application components that you have selected to monitor. |

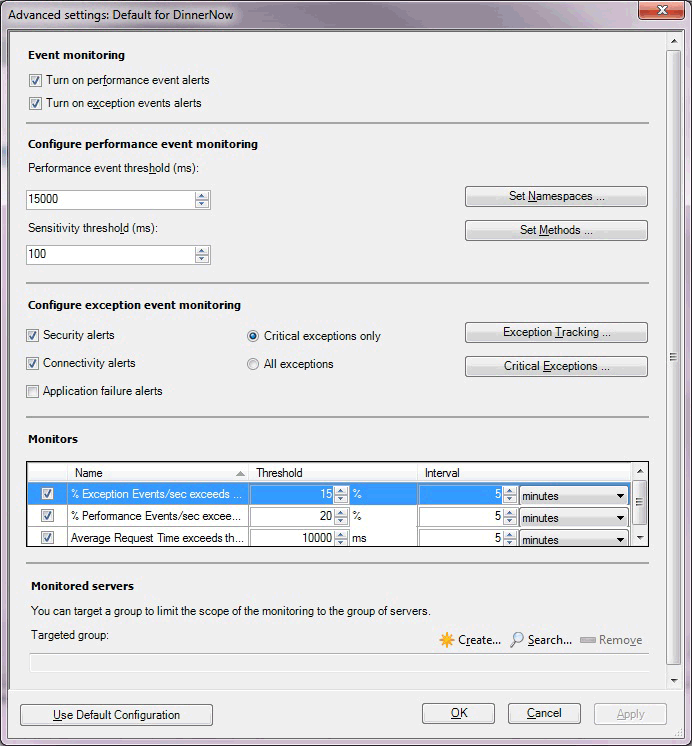
Server-Side Configuration



The following options are available on the Server-Side Configuration page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting for the application group on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn the exception event alert notification for the application group on or off within the Operations console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event. |
| Advanced Settings | Set advanced configurations, including sensitivity (restricting the collection of fast functions), namespaces (that define where you want to collect data from custom applications), methods (specific functions where you want to start monitoring), custom exception handlers (that define critical exceptions), and customize the configuration of the monitors that affect the component health state. |
| Enable additional configuration options for server-side and client-side monitoring | Specify additional options in the wizard to customize monitoring for individual application components and client-side monitoring. |

Advanced Settings for Server-Side Monitoring



The following options are available on the Advanced Settings for server-side monitoring page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting for the application group on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn the exception event alert notification for the application group on or off within the Operations console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a request must be processed in before it causes a performance event. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Set Namespaces | Specify namespaces and classes where to start measuring for performance events and performance threshold violations, and define which namespaces should be treated by default as entry points. For more information, see [How to Add, Enable, and Disable Namespaces](#z525024710857408fb17e73733225f65b) |
| Set Methods | Specify how deep in the call stack to drill down to collect more detailed information, such as parameters and variables, for specific methods. For more information, see [How to Add, Edit, and Remove Methods](#z84bf3d38dda849c7a7afc400835ab2e1) |
| Security alerts | Turn alerting of exceptions on or off that are classified as security alerts for the application group, with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Connectivity alerts | Turn alerting of exceptions on or off that are classified as connectivity alerts for the application group, with errors such as “Connection timed out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Application failure alerts | Turn alerting of exceptions on or off that are classified as application, or code, failures for the application group. By default, this option is turned off to reduce the “noise” of alerts raised due to code failures that typically only development teams can resolve. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Critical exceptions only | Specify whether the exception is considered relevant by the Application Performance Monitoring agent and whether an event is created. An event is created if the exception raised is one of those in the list of critical exception handlers. For more information, see [Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58) |
| All exceptions | Specify whether all exceptions are considered relevant by the Application Performance Monitoring agent and events are created when exceptions are detected in monitored namespaces and classes. |
| Exception Tracking | Select to add namespace or classes where you track exception parameters or variables, and collect additional information about each exception that a namespace or class raised. For more information, see [How to Add, Edit, and Remove Exception Tracking](#zc5cc4e3622594f5da5def96cf20545f5) |
| Critical Exceptions | Select to add items to the Exception handlers list. Define exception handlers that catch critical exceptions that an application raised. For more information, see [Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58) |
| Monitors: Exception Events/sec exceeds | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: Performance Events/sec exceeds | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time exceeds | Monitor that watches the .NET Apps/Average Request Time performance counter. |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

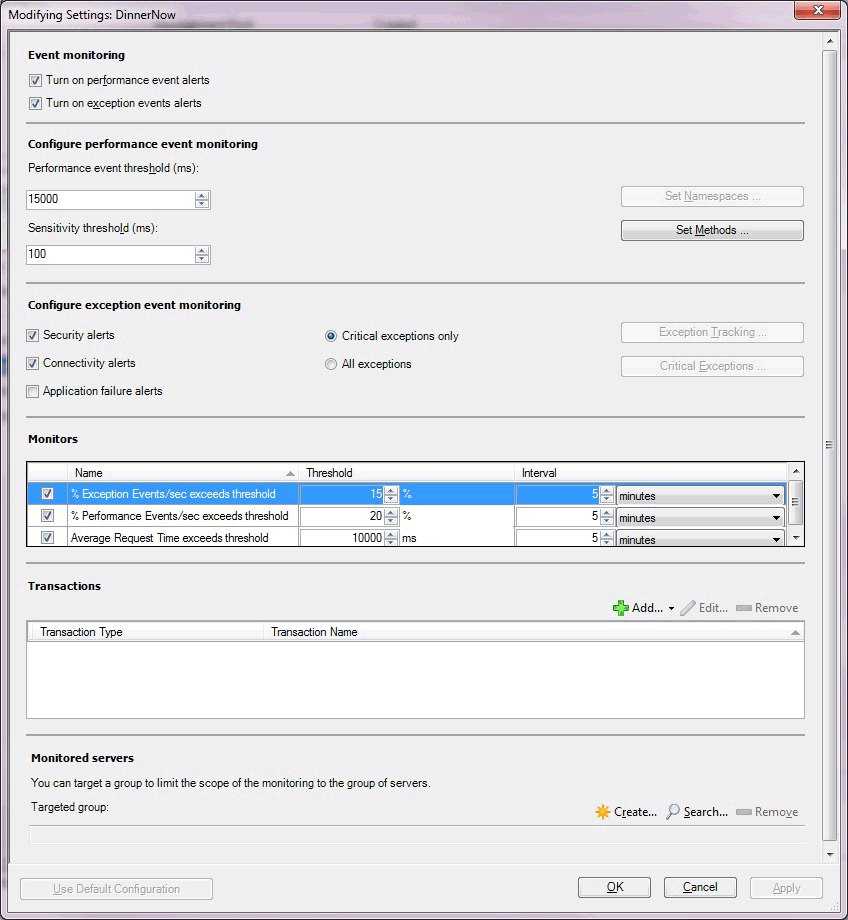
Server-Side Customization



For System Center 2012 SP1, the following options are available on the Server-Side Customization page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Component | Select the component you want to customize for monitoring individual application components. |
| Customize | Modify the settings for the selected application component. This opens the Modifying Settings page. The settings on this page are the same as those on the Advanced Settings for Server-Side Monitoring page, except you can create individual transaction monitoring for ASP.NET webpages, ASP.NET web services, or individual functions in an assembly. These are described in the [Transaction Properties: Add ASP.NET Web Page](#z31) sections that follow.  Note  The buttons for namespaces, exception tracking, and critical exceptions are unavailable because these can only be set at the application-group level, not at the component level. For System Center 2012 SP1 only: you can customize these settings if you are configuring monitoring for Windows Services. |
| Modifying Settings page | Customize settings for the application component and/or specify monitoring for a specific webpage, web method, or function within the application component. |

Server-Side Modifying Settings



The following options are available on the Server-Side Modifying Settings page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting for the application group on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn the exception event alert notification for the application group on or off within the Operations console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a request must be process in before it causes a performance event. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Set Methods | Specify how deep in the call stack to drill down to collect more detailed information, such as parameters and variables, for specific methods. For more information, see [How to Add, Edit, and Remove Methods](#z84bf3d38dda849c7a7afc400835ab2e1) |
| Security alerts | Turn on or off alerting of exceptions classified as security alerts for the application component, with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Connectivity alerts | Turn on or off alerting of exceptions classified as connectivity errors for the application component, such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Application failure alerts | Turn on or off alerting of exceptions classified as application, or code, failures for the application component. By default, this option is turned off to reduce the “noise” of alerts raised due to code failures that typically only development teams can resolve. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Critical exceptions only | Specify whether the exception is considered relevant by the Application Performance Monitoring agent and whether an event is created. An event is created if the exception raised is one of those in the list of critical exception handlers. For more information, see [Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58) |
| All exceptions | Specify whether all exceptions are considered relevant by the Application Performance Monitoring agent and events are created when exceptions are detected in monitored namespaces and classes. |
| Monitors: Exception Events/sec exceeds | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: Performance Events/sec exceeds | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time exceeds | Monitor that watches the .NET Apps/Average Request Time performance counter. |
| Transactions: Add | Add transactions for ASP.NET web pages, ASP.NET Web services, and functions. See the following tables. |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

Note

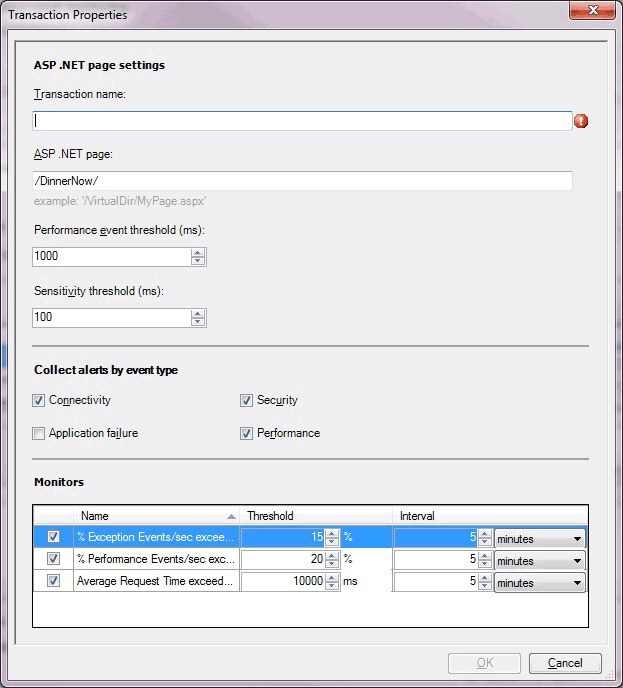
The buttons for namespaces, exception tracking, and critical exceptions are unavailable because these can only be set at the application-group level, not at the component level. For System Center 2012 SP1 only: You can customize these settings if you are configuring monitoring for Windows services.

Application types and server-side transactions you can monitor

For each application type there are several transaction types you can choose to monitor. The following options are available:

|  |  |  |
| --- | --- | --- |
| Application type | Transaction types for System Center 2012 | Transaction types for System Center 2012 SP1 |
| ASP.NET Web application | **** ASP.NET webpage  **** ASP.NET web service  **** Function | **** ASP.NET webpage  **** ASP.NET MVC page  **** ASP.NET web service  **** WCF method  **** Function |
| ASP.NET Web service | **** ASP.NET webpage  **** ASP.NET web service  **** Function | **** ASP.NET webpage  **** ASP.NET MVC page  **** ASP.NET web service  **** WCF method  **** Function |
| WCF service | Not available | **** ASP.NET webpage  **** ASP.NET MVC page  **** ASP.NET web service  **** WCF method  **** Function |
| Windows service | Not available | **** WCF method  **** Function |

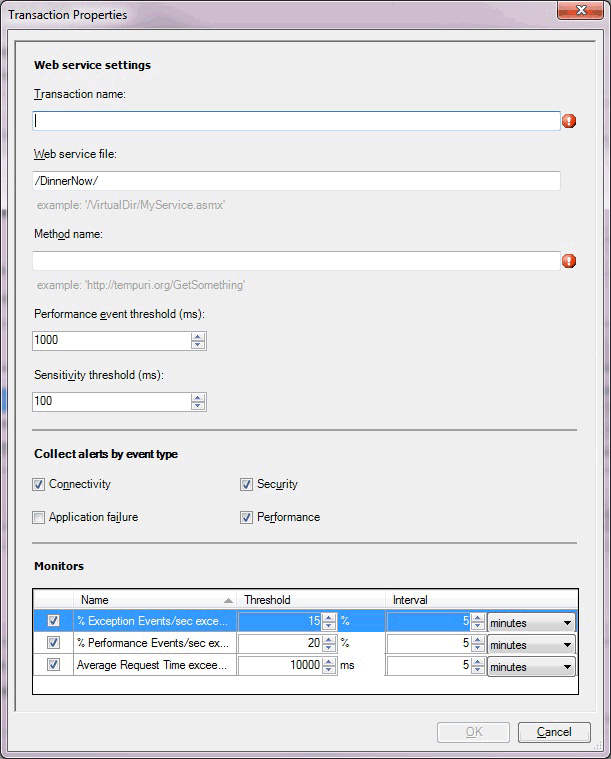
Transaction Properties: Add ASP.NET Web Page



The following options are available on the Transaction Properties page for ASP.NET Web Page page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Transaction name | Enter the friendly name for the transaction as it will be displayed on the Monitoring tab, performance counters, and elsewhere. |
| ASP.NET page | Enter the path to the page that you are configuring these monitoring settings for. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event.  Note  The application component continues to monitor the page specified in the transaction by using the performance threshold that is set for the application component. This threshold is used as a second measure on the same page in the application component. If you set this threshold higher than the application component threshold, you get a single event, but you might get two performance alerts for the transaction when the threshold is breached—one from the application component and one from the transaction, depending on your alerting settings. Transactions are typically used to monitor the individual page more aggressively than the parent application, at a lower threshold, or to monitor a page where alerting has been disabled on the parent. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. An event is still generated if the threshold is surpassed. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Collect alerts by event type: Connectivity | Turn on or off alerting of events, classified as connectivity alerts with errors such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Application failure | Turn on or off alerting of events classified as application, or code, failures. Turning this off reduces the “noise” of many alerts raised due to code failures. Because these alerts are raised from code failures, developers usually resolve these issues. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Security | Turn on or off alerting of events classified as security alerts, with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Performance | Turn on or off alerting of events classified as performance alerts.  Performance events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Monitors: % Exception Events/sec | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: % Performance Events/sec | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time | Monitor that watches the .NET Apps/Average Request Time performance counter. |

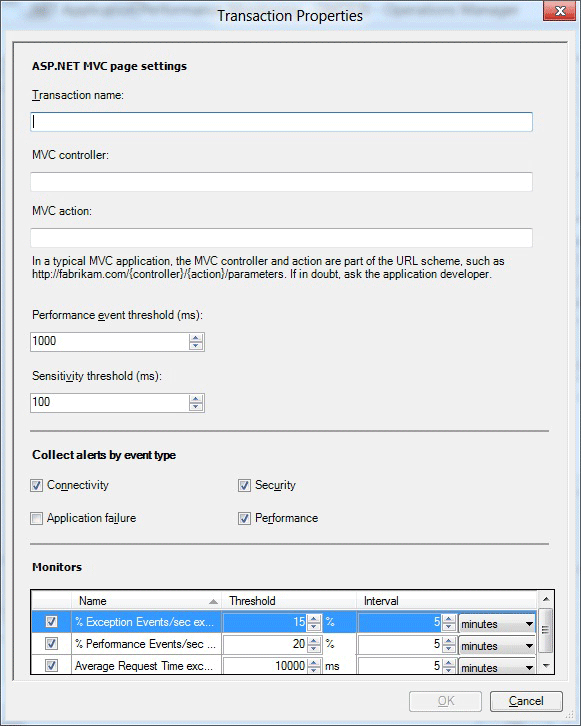
Transaction Properties: Add ASP.NET Web Service



The following options are available on the Transaction Properties page for the ASP.NET Web Service page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Transaction name | Enter the friendly name for the transaction as it will be displayed on the Monitoring tab, performance counters, and so on. |
| Web service file | Enter the path to the file for which you are configuring these monitoring settings. |
| Method name | Enter the URI of the web method that you want to monitor. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event.  Note  The application component continues to monitor the page specified in the transaction by using the performance threshold that is set for the application component. This threshold is used as a second measure on the same page in the application component. If you set this threshold higher than the application component threshold, you get a single event, but you might get two performance alerts for the transaction when the threshold is breached—one from the application component and one from the transaction, depending on your alerting settings. Transactions are typically used to monitor the individual page more aggressively than the parent application, at a lower threshold, or to monitor a page where alerting has been disabled on the parent. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Collect alerts by event type: Connectivity | Turn on or off alerting of events classified as connectivity alerts, with errors, such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Application failure | Turn on or off alerting of events classified as application, or code, failures. Turning this option off reduces the “noise” of many alerts raised due to code failures. Because these alerts are raised from code failures, developers usually resolve these issues. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Security | Turn on or off alerting of events classified as security alerts, with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Performance | Turn on or off alerting of events classified as performance alerts.  Performance events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Monitors: % Exception Events/sec | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: % Performance Events/sec | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time | Monitor that watches the .NET Apps/Average Request Time performance counter. |

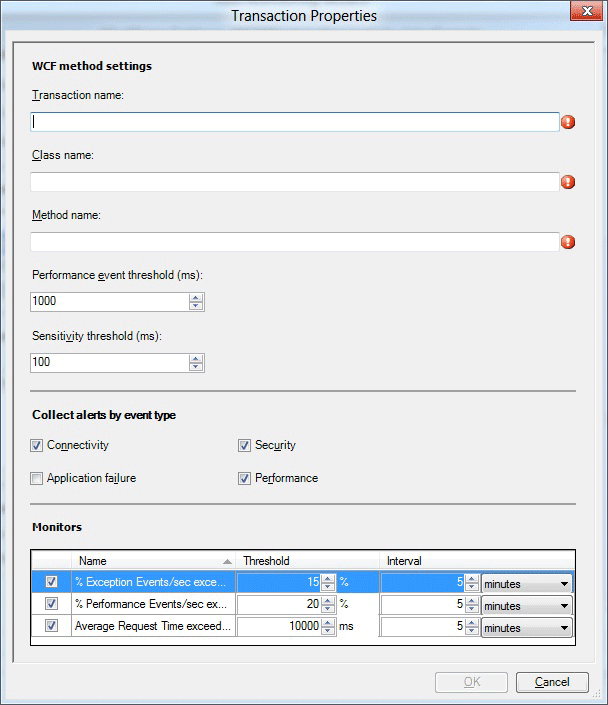
Transaction Properties: Add ASP.NET MVC Page



For System Center 2012 SP1 the following options are available on the Transaction Properties for the ASP.MVC page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Transaction name | Enter the friendly name for the transaction as it will be displayed on the Monitoring tab, performance counters, and so on. |
| MVC controller | Enter the name of the MVC controller for which you are configuring these monitoring settings. |
| MVC action | Specify the name of the MVC action for which you are configuring these monitoring settings. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event.  Note  The application component continues to monitor the page specified in the transaction by using the performance threshold that is set for the application component. This threshold is used as a second measure on the same page in the application component. If you set this threshold higher than the application component threshold, you get a single event, but you might get two performance alerts for the transaction when the threshold is breached—one from the application component and one from the transaction, depending on your alerting settings. Transactions are typically used to monitor the individual page more aggressively than the parent application, at a lower threshold or to monitor a page where monitoring has been disabled on the parent. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Collect alerts by event type: Connectivity | Turn on or off alerting of events classified as connectivity alerts, with errors such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Application failure | Turn on or off alerting of events classified as application, or code, failures. Turning this option off reduces the “noise” of many alerts raised due to code failures. Because these alerts are raised from code failures, developers usually resolve these issues. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Security | Turn on or off alerting of events classified as security alerts with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Performance | Turn on or off alerting of events classified as performance alerts.  Performance events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Monitors: % Exception Events/sec | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: % Performance Events/sec | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time | Monitor that watches the .NET Apps/Average Request Time performance counter. |

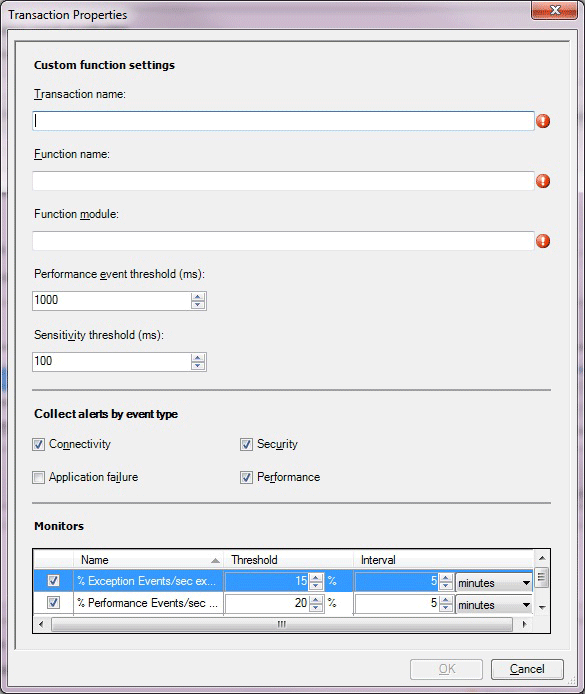
Transaction Properties: Add WCF Method



The following options are available on the Transaction Properties for the Add WCF method settings page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Transaction name | Enter the friendly name for the transaction as it will be displayed on the Monitoring tab, performance counters, and so on. |
| Class name | Enter the name of the class for which you are configuring these monitoring settings. The class name is in the format: Namespace.Class. For example: wcfservice.myclass. |
| Method name | Specify the name of the method that is expected to be in the class for which you are configuring these monitoring settings. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event.  Note  The application component continues to monitor the page specified in the transaction by using the performance threshold that is set for the application component. This threshold is used as a second measure on the same page in the application component. If you set this threshold higher than the application component threshold, you get a single event, but you might get two performance alerts for the transaction when the threshold is breached—one from the application component and one from the transaction, depending on your alerting settings. Transactions are typically used to monitor the individual page more aggressively than the parent application, at a lower threshold or to monitor a page where alerting has been disabled on the parent. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Collect alerts by event type: Connectivity | Turn on or off alerting of events classified as connectivity alerts, with errors such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Application failure | Turn on or off alerting of events classified as application, or code, failures. Turning this option off reduces the “noise” of many alerts raised due to code failures. Because these alerts are raised from code failures, developers usually resolve these issues. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Security | Turn on or off alerting of events classified as security alerts with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Performance | Turn on or off alerting of events classified as performance alerts.  Performance events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Monitors: % Exception Events/sec | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: % Performance Events/sec | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time | Monitor that watches the .NET Apps/Average Request Time performance counter. |

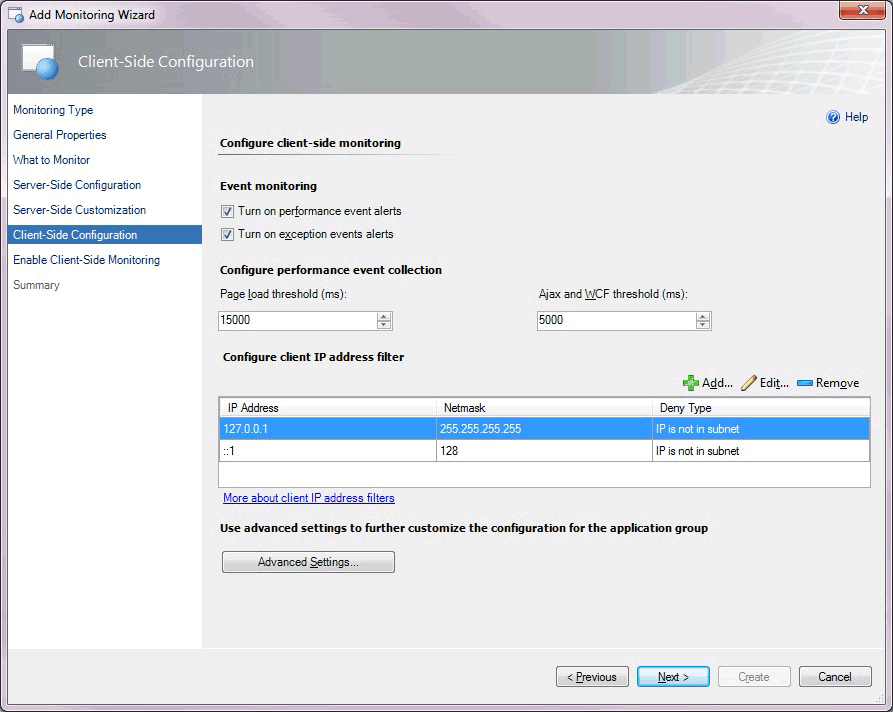
Transaction Properties: Add Function



The following options are available on the Transaction Properties for the Add Function page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Transaction name | Enter the friendly name for the transaction as it will be displayed on the Monitoring tab, performance counters, and so on. |
| Function name | Enter the name of the function for which you are configuring these monitoring settings. The function name is in the format: Namespace.Class.Method. For example: System.Web.UI.Page.ProcessRequest. |
| Function module | Specify the name of the assembly, such as System.Web.dll, that defines the function for which you are configuring these monitoring settings. |
| Performance event threshold (ms) | Set the threshold in milliseconds (ms) that a user transaction must exceed before it raises a performance event.  Note  The application component continues to monitor the page specified in the transaction by using the performance threshold that is set for the application component. This threshold is used as a second measure on the same page in the application component. If you set this threshold higher than the application component threshold, you get a single event, but you might get two performance alerts for the transaction when the threshold is breached—one from the application component and one from the transaction, depending on your alerting settings. Transactions are typically used to monitor the individual page more aggressively than the parent application, at a lower threshold or to monitor a page where alerting has been disabled on the parent. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Collect alerts by event type: Connectivity | Turn on or off alerting of events classified as connectivity alerts, with errors such as “Connection Timed Out”.  Connectivity events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Application failure | Turn on or off alerting of events classified as application, or code, failures. Turning this option off reduces the “noise” of many alerts raised due to code failures. Because these alerts are raised from code failures, developers usually resolve these issues. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)  Exception events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Security | Turn on or off alerting of events classified as security alerts with errors such as “Access Denied” or “Login Failed”.  Security events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Collect alerts by event type: Performance | Turn on or off alerting of events classified as performance alerts.  Performance events are logged to the Application Diagnostics console. You have the option to choose whether to raise alerts after an Application Performance Monitoring event is generated. |
| Monitors: % Exception Events/sec | Monitor that watches the .NET App/% Exception Events/sec performance counter. |
| Monitors: % Performance Events/sec | Monitor that watches the .NET Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time | Monitor that watches the .NET Apps/Average Request Time performance counter. |

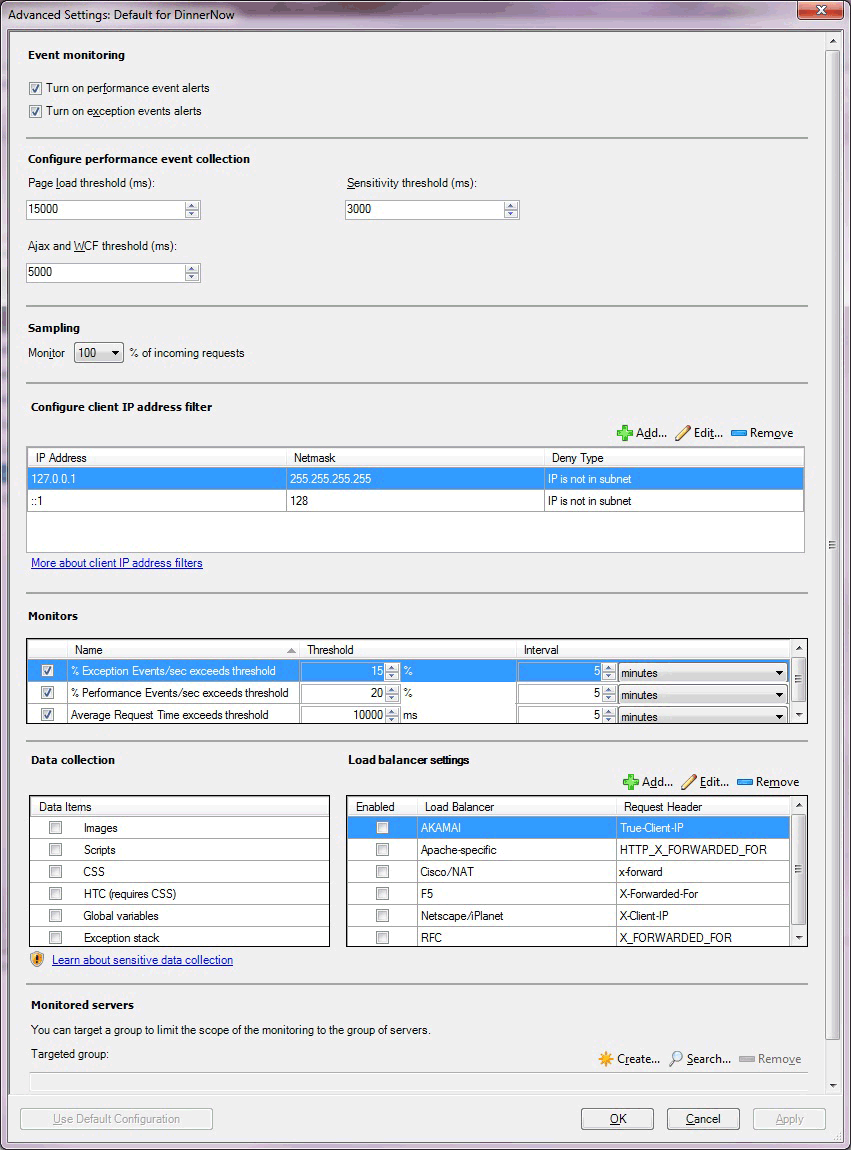
Client-Side Configuration



The following options are available on the Client-Side Configuration page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn exception event alert reporting on or off within the Operations Manager console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Page load threshold (ms) | Set the threshold in milliseconds (ms) that a page load must exceed before it causes a performance event. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. The event is only turned into an alert if you have selected Turn on performance event alerts. |
| IP address filter: IP Address | Specify the IP addresses that you want to exclude from monitoring. For more information, see [How to Configure IP Address Exclusion Filters for Client-Side Monitoring](#z0529d193725b48bfbfcceb153985e298) |
| IP address filter: Netmask | The part of the filter IP address and user IP address that have to be compared for equality. |
| IP address filter: Comparison Type | Specify to exclude IP addresses that match the IP addresses in the subnet (IP is in subnet), or to exclude the user IP addresses that do not match the IP addresses in the subnet (IP is not in subnet). |
| IP address filter: Use IPv6 | Add the IPv6 filter if the IPv6 protocol is enabled on the web server. |
| Advanced Settings | Specify settings, such as performance and event monitoring thresholds, exception event monitoring, Critical Exceptions, and monitors. |

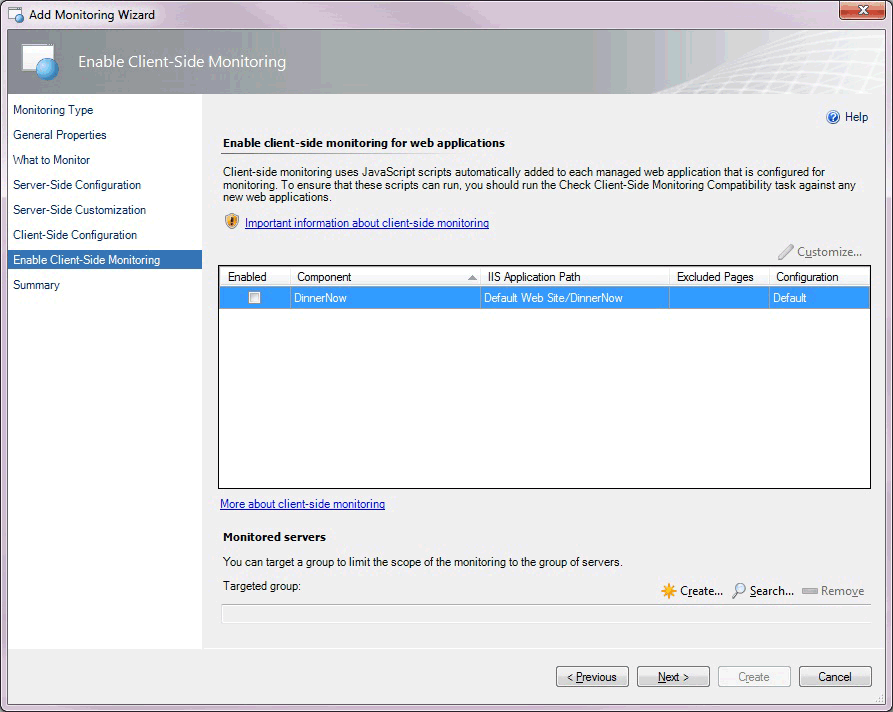
Advanced Settings for Client-Side Monitoring



The following options are available on the Advance Settings for Client-Side Monitoring page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn exception event alert notification on or off within the Operations console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Page load threshold (ms) | Set the threshold in milliseconds (ms) that a page load must exceed before it causes a performance event. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. The event is only turned into an alert if you have selected Turn on performance event alerts. |
| Ajax and WCF threshold (ms) | Set the threshold in milliseconds (ms) that an Ajax or Windows Communications Foundation (WCF) call initiated from the page must exceed before it causes a performance event. The event is only turned to an alert if you have selected Turn on performance event alerts. |
| Monitor % of incoming requests. | Specify a sample size of incoming requests, defined as a percentage of the total number of incoming requests that you want to monitor. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| IP address: IP Address | Specify the IP addresses that you want to exclude from monitoring. For more information, see [How to Configure IP Address Exclusion Filters for Client-Side Monitoring](#z0529d193725b48bfbfcceb153985e298) |
| IP address: Netmask | Specify the part of the filter IP address and user IP address that have to be compared for equality. |
| IP address: Comparison Type | Specify to exclude IP addresses that match the IP addresses in the subnet (IP is in subnet), or to exclude the user IP addresses that do not match the IP addresses in the subnet (IP is not in subnet). |
| IP address: Use IPV6 | Specify to add the IPv6 filter if the IPv6 protocol is enabled on the web server. |
| Monitors: Exception Events\sec exceeds | Monitor that watches the .NET CSM Apps/% Exceptions Events/sec performance counter. |
| Monitors: Performance Events\sec exceeds | Monitor that watches the .NET CSM Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time exceeds | Monitor that watches the .NET CSM Apps/Average Page Load Response Time performance counter. |
| Data Items | Select the type of client-side data that you want to collect. For more information, see [Working with Sensitive Data for .NET Applications](http://go.microsoft.com/fwlink/?LinkId=231757) |
| Load balancer settings | Select the type of load balancer that you are using with your application. You can also add your own load balancer, if it is not included in the list. For more information, see [Client-Side Monitoring with Targeted Groups and Load Balancers](#z474bdebf6bbc429fa791a382d95113bc) |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

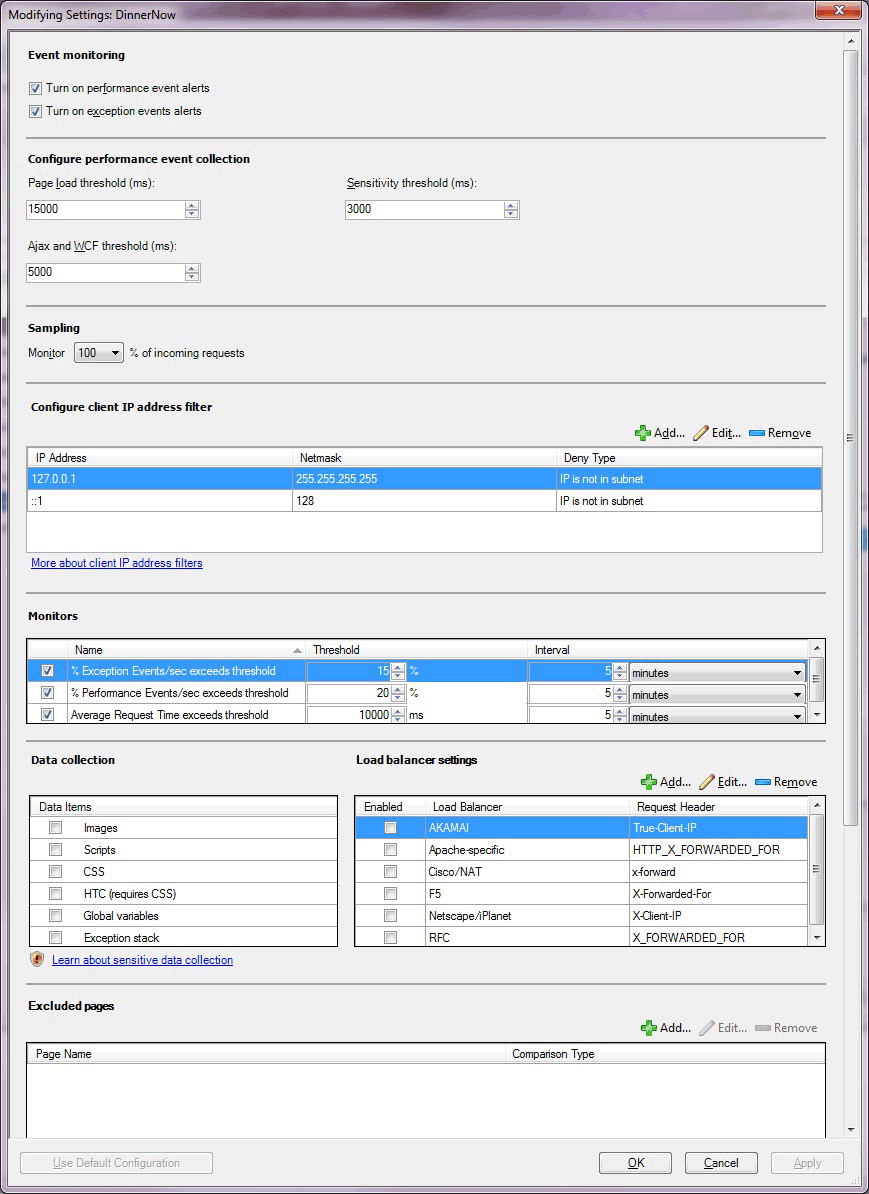
Enable Client-Side Monitoring



The following options are available on the Enable Client-Side Monitoring page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Component | Select the component you want to customize for monitoring individual application components. Only the components of the ASP.NET Web application type are displayed. Web Services and WCF Services do not serve HTML pages to browsers, so you cannot enable client-side monitoring for them. For System Center 2012 SP1 only: .NET applications hosted in Windows Services do not serve HTML pages to browsers, so you cannot enable client-side monitoring for them. |
| Customize | Modify the settings for the selected application component. This opens the Modifying Settings page. The settings on this page are similar to those on the Advanced Settings for Client-Side Monitoring page. On the Modifying Settings page, you can specify the pages to be excluded from monitoring. |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

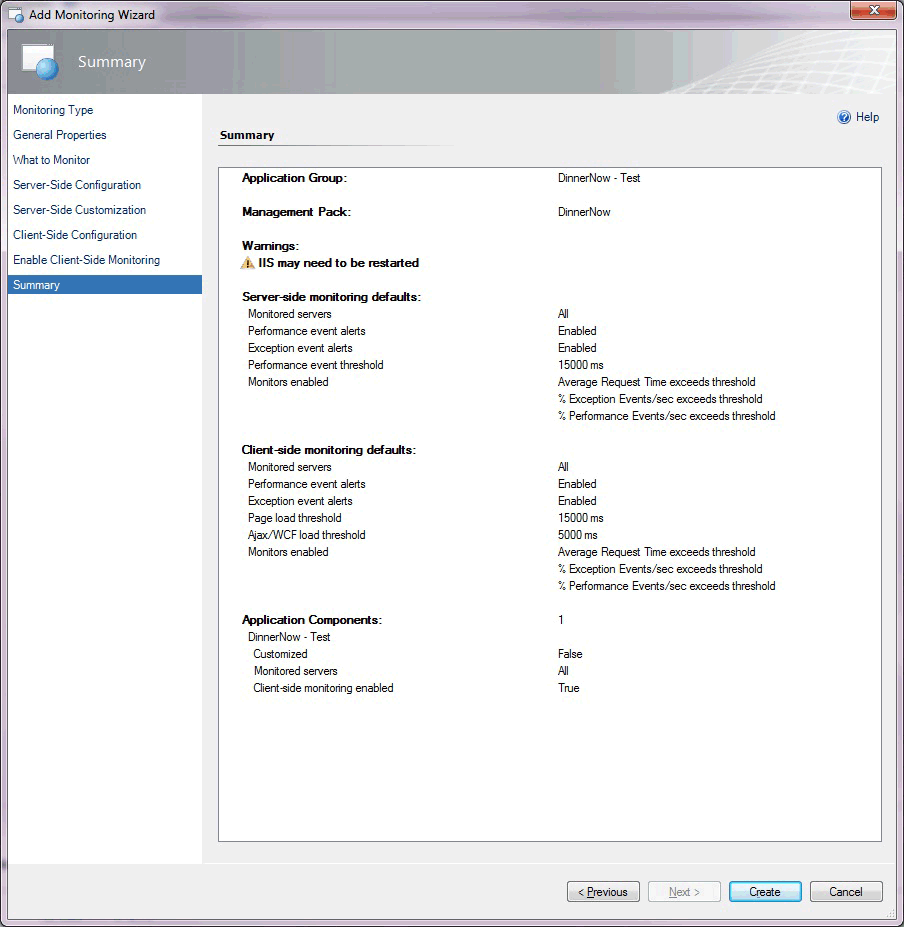
Client-Side Modifying Settings



The following options are available on the Client-Side Modifying Settings page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Turn on performance event alerts | Turn performance event alert reporting on or off within the Operations console for server-side monitoring. Performance events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Turn on exception event alerts | Turn exception event alert reporting on or off within the Operations console for server-side monitoring. Exception events are still logged to the Application Diagnostics console. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. |
| Page load threshold (ms) | Set the threshold in milliseconds (ms) that a page load must exceed before it causes a performance event alert. You have the option whether to raise alerts after an Application Performance Monitoring event is generated. The event is only turned into an alert if you have selected Turn on performance event alerts. |
| Ajax and WCF threshold (ms) | Sets the threshold in milliseconds that an Ajax or Windows Communications Foundation (WCF) call initiated from the page must exceed before it causes a performance event. The event is only turned into an alert if you have selected Turn on performance event alerts. |
| Sensitivity threshold (ms) | Specify to filter out fast-running methods to reduce overall “noise” by reducing the size of the call stack by gathering less data for each event. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| Monitor % of incoming requests. | Specify a sample size of incoming requests, defined as a percentage of the total number of incoming requests that you want to monitor. For more information, see [Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830) |
| IP address: IP Address | Enter the IP addresses that you want to exclude from monitoring. For more information, see [How to Configure IP Address Exclusion Filters for Client-Side Monitoring](#z0529d193725b48bfbfcceb153985e298) |
| IP address: Netmask | Specify the part of the filter IP address and user IP address that have to be compared for equality. |
| IP address: Comparison Type | Specify to exclude IP addresses that match the IP addresses in the subnet (IP is in subnet), or to exclude the user IP addresses that do not match the IP addresses in the subnet IP is not in subnet). |
| IP address: Use IPV6 | Specify to add the IPv6 filter if the IPv6 protocol is enabled on the web server. |
| Monitors: Exception Events\sec exceeds | Monitor that watches the .NET CSM Apps/% Exceptions Events/sec performance counter. |
| Monitors: Performance Events\sec exceeds | Monitor that watches the .NET CSM Apps/% Performance Events/sec performance counter. |
| Monitors: Average Request Time exceeds | Monitor that watches the .NET CSM Apps/Average Page Load Response Time performance counter. |
| Data collection | Select the type of client-side data you want to collect. For more information, see [Working with Sensitive Data for .NET Applications](http://go.microsoft.com/fwlink/?LinkId=231757) |
| Load balancer settings | Select the type of load balancer that you are using with your application. You can also add your own load balancer, if it is not included in the list. For more information, see [Client-Side Monitoring with Targeted Groups and Load Balancers](#z474bdebf6bbc429fa791a382d95113bc). |
| Excluded pages: Add | Specify to add the pages to exclude from monitoring. You typically exclude pages that are considered unimportant for given metrics or that did not pass the compatibility check. |
| Transactions: Add | Specify transactions to add for ASP.NET web pages. |
| Targeted group | Select specific servers to limit monitoring to this specific set of servers. This is optional.  Targeted group scoping only becomes necessary when you have the same application running in multiple environments, such as production and staging, and you intend to run the template multiple times, one for each environment. In this scenario, group which machines belong to production and which belong to the staging environment, and then use the targeted groups to restrict where the configuration is propagated. You can also use groups to apply configuration to a subset of your servers. Otherwise, it is not necessary to specify targeted group scoping if you just want to monitor all instances of a given application. |

Summary



The Summary page of the wizard lists the settings you have configured for the .NET Application Performance Monitoring template. If you want to change any of these settings, click Previous or the template page until you reach the page with the settings that you want to change.

Creating and Modifying .NET Application Performance Monitoring Templates

For the procedure to run the .NET Application Performance Monitoring wizard, see [How to Configure Monitoring for .NET Applications](#z18139deace8442b8bd50d467d72623e1).

To modify an existing .NET Application Performance Monitoring template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials in the management group.  2. Click the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select .NET Application Performance Monitoring.  4. In the .NET Application Performance Monitoring pane, locate the template to change.  5. Right-click the application group that you want to modify, and then select Properties.  6. Using the tabs to navigate the pages of settings, make the desired changes, such as adding customized monitoring for a specific application component or configuring and enabling client-side monitoring, and then click OK. |

Viewing .NET Application Performance Monitoring Monitors and Collected Data

After you configure monitoring for an application, these three views will help you get started with the monitoring experience.

To view all .NET Application Performance Monitoring monitored applications

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand .NET Monitoring, and then click Monitored Applications. |

To view the state of each monitor

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand .NET Monitoring, and then click Monitored Applications.  4. In the Monitored Applications view, right-click an object. Select Open, and then click Health Explorer.  5. Expand the Availability and Performance nodes to view the individual monitors. |

To view the performance collected for an application component

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand .NET Monitoring, and then click Monitored Applications.  4. In the Monitored Applications pane, right-click an object. Select Open, and then click Performance View.  5. In the Legend pane, select the counters that you want to view.  6. Use options in the Actions pane to modify the Performance view. |

See Also

[Before You Begin Monitoring .NET Applications](#z43beed71594a40d9aa347210b0e73bcb)

[How to Configure Monitoring for .NET Applications](#z18139deace8442b8bd50d467d72623e1)

[How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9)

[Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)

Before You Begin Monitoring .NET Applications

In System Center 2012 – Operations Manager, you can monitor web applications and web services from server- and client-side perspectives to get details about application availability and performance that can help you pinpoint problems. (For System Center 2012 SP1 only: You can also monitor Windows Services.) When you specify settings, the types of events to collect, the performance goals to measure, and which servers to monitor, Operations Manager .NET Application Monitoring provides insights into how web-based applications are running. You can see how frequently a problem is occurring, how a server was performing when a problem occurred, and the chain of events related to the slow request or a method that is unreliable. You have to have this information to partner with software developers and database administrators to help ensure that applications are available and perform at optimal levels.

Current AVIcode 5.7 customers: Operations Manager has .NET application monitoring capabilities that can replace AVIcode 5.7 for applications hosted in Internet Information Services (IIS) 7.0. (For System Center 2012 SP1 only: Also for applications hosted in Internet Information Services (IIS) 8.0.) If you are using AVIcode 5.7 to monitor applications and want to upgrade to Operations Manager, you must be running AVIcode 5.7 with System Center Operations Manager 2007 R2 with the latest cumulative updates. During the upgrade, the AVIcode 5.7 configuration is not converted to the .NET Application Performance Monitoring configuration, and the AVIcode agent is not replaced with the .NET Application Performance Monitoring agent. AVIcode agents continue to work as they did before the upgrade because the .NET Application Performance Monitoring configuration does not affect them. However, to continue to monitor applications by using AVIcode 5.7 after the upgrade, you must manually install these management packs: AVIcode.DotNet.SystemCenter.Enterprise.Monitoring.mpb and AVIcode.DotNet.SystemCenter.Client.Monitoring.mp. For more information, see [Steps to import AVIcode 5.7 templates after upgrading](http://go.microsoft.com/fwlink/?LinkId=230859) and [Notes for AVIcode 5.7 Customers](http://go.microsoft.com/fwlink/?LinkId=231263)

After the upgrade, we recommend that you configure all new application monitoring by using the .NET Application Performance Monitoring wizard and gradually replace the AVIcode 5.7 agents with the new .NET Application Performance Monitoring agents.

Before You Begin

To view Application Performance Monitoring event details, you must install the Operations Manager Web Console. For more information, see [How to Install the Operations Manager Web Console](http://go.microsoft.com/fwlink/?LinkId=236163)

Import the IIS 7.0 management pack (Microsoft.Windows.InternetInformationServices.2008.mp) and the Operations Manager APM Web IIS 7 management pack (Microsoft.SystemCenter.Apm.Web.IIS7.mp). The Microsoft.SystemCenter.Apm.Web.IIS7.mp management pack is included in the download package in the Management Packs folder—it is not available on the management pack catalog.

For System Center 2012 SP1, to monitor Windows Server 2012, you also need to import the Microsoft.InternetInformationServices.2012.mp management pack (in the management pack catalog) and the Microsoft.SystemCenter.Apm.Web.IIS8.mp management pack. The Microsoft.SystemCenter.Apm.Web.IIS8 mp management pack is included in the download package in the Management Packs folder—it is not available on the management pack catalog.

For information about importing management packs, see How to Import an Operations Manager Management Pack. Additionally, ensure that websites, applications, and services have been discovered.

Important

When you try to import the .NET Monitoring management pack (Microsoft.SystemCenter.Apm.Web.IIS7.mp) from disc and resolve the references it has to Microsoft.Windows.InternetInformationServices.2008.mp, this IIS management pack in turn references the Windows Server 2008 Operating System (Discovery) management pack. If any of these management packs fail to import manually, update Microsoft.Windows.Server.Library.mp and import Microsoft.SystemCenter.Apm.Web.IIS7.mp again.

If you are running Windows Server 2008 R2, import the IIS 7.0 management packs.

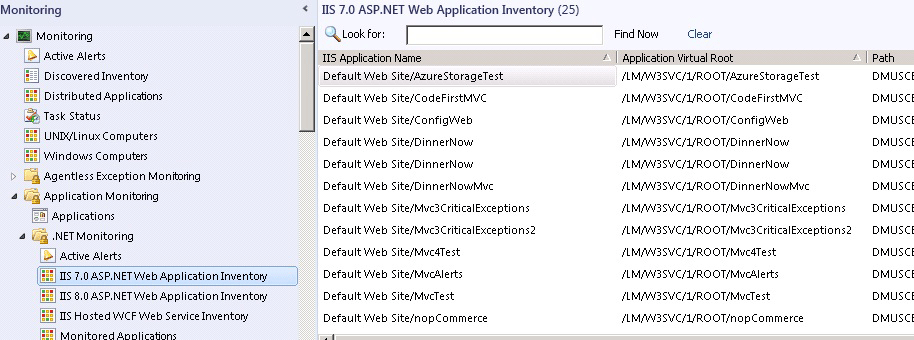
For System Center 2012 SP1 only: If you are running Windows Server 2012, import the IIS 8.0 management pack. If you are running both Windows Server 2008 R2 and Windows Server 2012, import IIS 7.0 and IIS 8.0 management packs.

Note

After you import the management packs, wait until the IIS Application Pools have been discovered and ASP.NET Web Applications start to appear in the ASP.NET Web Application Inventory view before you begin monitoring. For more information, see [Configuring the Management Pack for Operations Manager APM Web IIS 7](http://go.microsoft.com/fwlink/?LinkId=251495)

Before you begin monitoring Windows Services, you need to configure Windows Services using the Windows Service template. Once you do this, the.NET Application Performance Monitoring template can discover the Windows Services that are running. For more information, see [Authoring the Windows Service Template](http://go.microsoft.com/fwlink/?LinkId=252385)

ASP.NET Web Application Inventory view



Here are some questions to answer before you begin:

 What are the names of the applications that you want to monitor?

 Which servers host the applications that you want to monitor?

 Do you want to monitor performance events?

 Do you want to monitor exception events?

 What threshold signifies a performance issue with a request?

 Do you want to monitor both server and browser performance?

If you plan to monitor browser performance, check application compatibility before you begin to configure client-side monitoring of the ASP.NET application. To check compatibility, use the Check Client-Side Compatibility task. The information collected by this task tells you what pages should be excluded from monitoring. The list of excluded pages is entered during the configuration of client-side application monitoring in the wizard.

To check applications for compatibility with client-side monitoring

|  |
| --- |
| 1. To see a view of all the ASP.NET web applications that Operations Manager finds, in the Operations console, in the navigation pane, click the Monitoring button, expand Application Monitoring, expand .NET Monitoring, and then click ASP.NET Web Application Inventory.  2. To check applications for compatibility with client-side monitoring, click an application name, and in the tasks pane, in the IIS 7 ASP.NET Application Endpoint Tasks section, click Check Client Side Monitoring Compatibility, and then click Run. This task does an analysis and lets you know if the application has compatibility issues with client-side monitoring.  3. The results page displays whether incompatibilities are found. If an application has incompatibilities, you should exclude the incompatible pages from client-side monitoring later when you are configuring client-side monitoring.  You can change an application so that it becomes compatible with client-side monitoring by following the instructions in the output of the compatibility tool. |

Note

For System Center 2012 SP1, you can only check applications for compatibility with client-side monitoring in the IIS 7.0 ASP.NET Web Applications and IIS 8.0 ASP.NET Web Applications views. You cannot check applications for compatibility with client-side monitoring in the IIS Hosted WCF Web Services view because client-side monitoring does not apply to this type of endpoint.

See Also

[How to Configure Monitoring for .NET Applications](#z18139deace8442b8bd50d467d72623e1)

[How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9)

How to Configure Monitoring for .NET Applications

Configure .NET Application Performance Monitoring

The .NET Application Performance Monitoring template in System Center 2012 – Operations Manager lets you monitor .NET and WCF applications hosted in Internet Information Services (IIS) 7.0. For System Center 2012 SP1 only: You monitor applications hosted in IIS 8.0 and Windows Services. You can select one or more applications or services and configure monitoring of performance and exception events. Server-side monitoring lets you measure details about the performance and reliability of applications that are running in your datacenter. By monitoring client-side applications, you can measure details of the customer experience, such as how long it takes for a page to load. It is another way to monitor how your applications are working from the perspective of your customer. Client-side application monitoring helps you determine whether your users are experiencing problems. With both client-side and server-side monitoring in use, you can determine if a problem exists on your server, in the application, or is being caused by external factors, such as high network latency.

Tip

Client-side monitoring can be set up at the same time as server-side monitoring when you run the .NET Application Performance Monitoring wizard or by editing an existing instance of a template.

Important

You can only configure client-side monitoring for applications that have been configured for server-side monitoring.

To configure .NET Application Performance Monitoring (server-side perspective)

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, click .NET Application Performance Monitoring, and then, in the tasks pane, click the Add Monitoring Wizard where you name and configure the application group that you want to monitor.  Location of .NET Application Performance Monitoring    2. In the Add Monitoring Wizard on the Monitoring Type page, select .NET Application Performance Monitoring, and then click Next. This template lets you monitor web applications and services hosted in IIS 7.0. (For System Center 2012 SP1 only: You can monitor applications hosted in IIS 8.0 and Windows Services.) You can select one or more applications or services discovered by the IIS 7.0 management pack and configure monitoring of performance and exception events. For System Center 2012 SP1 only: You can select one or more applications or services discovered by the IIS 8.0 management pack or Windows Services previously configured with the Windows Service Template.  3. On the General Properties page, enter a friendly name and description for the application group that you are creating.  In the Select destination management pack menu, select the management pack to store the settings that are specific to this instance of the template. To create a new management pack, click New. In the Create a Management Pack wizard, name your new management pack the same as the application group so you can easily pair the two, which is helpful later in the monitoring experience. Click Next. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  4. On the What to Monitor page, in the Application components section, click Add. On the Object Search page, on the Search for menu, use the Filter by part of the name (optional) box to narrow your search, and then click Search to view a list of the application components you can monitor. (For System Center 2012 SP1 only: You can monitor Windows Services.) From the search results, select the application components that you want to monitor, click Add, and then click OK. The application components you selected are now displayed as members of the application group that you are going to monitor. Click Next.  On the What to Monitor page, on the Environment menu, select the environment you want to monitor your application in: None, Production, Staging, Test, Development, or New. Typically, you want to pair the environment tag with the server group you are monitoring.  Tip  If you do not have to monitor multiple versions of the same applications, such as production instances and staging instances, you can leave the environment tag set to None.  5. To limit the scope of monitoring to a group of servers, on the What to Monitor page, in the Monitored Servers section, click Search. On the Group Search page that opens, select the Filter by box and Management pack menu to find the server group that you want to use, and then click Search. Select the server group to which you want to limit monitoring in the Available Groups search results list, and click OK to add it to your targeted server group to monitor. Click Next.  Tip  The Targeted servers group lets you configure monitoring by using one set of thresholds for one set of application servers and a different set of thresholds for another set of application servers. To configure monitoring for the second set of application servers, run the template again, and use the alternate Targeted servers group and use a different environment tag for each template instance.  6. On the Server-Side Configuration page, decide how you want to configure your monitoring. You have options to:   Turn on or off performance event monitoring   Turn on or off exception event monitoring   Change the Performance Event Threshold   Configure Advanced Settings   Enable additional configuration options for server-side and client-side monitoring.  7. To further configure exception and performance event monitoring for the application group, including settings for Namespaces, Methods, Exception Tracking, and Critical Exception Handlers, click Advanced Settings. Also on the Advanced Settings for Server-Side Monitoring page, you can reset monitor thresholds from the defaults and scope monitoring to a targeted group. If you want to use or return to the default Advanced Settings, click Use Default Configuration. When you are finished, click OK. For more information, see [How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9) and [Application Monitoring Using the Default Settings](#zabd195c1584a4eac9c4554ede7d36845)  Warning  Gathering detailed performance and exception events can lead to collecting sensitive information that should not be passed on to the development team. For example, if you capture an exception from your billing system, you might also capture user names and other tokens that can be used to identify the person who is having problems making purchases and what they were trying to purchase. Before enabling the collection of parameters and local variables for performance and exception events, we recommend that you review your policies. For more information, see [Working with Sensitive Data for .NET Applications](http://go.microsoft.com/fwlink/?LinkId=231757)  8. If you only want to configure server-side monitoring and do not want to customize additional server-side monitoring options or configure and enable client-side monitoring, click Next, and on the Summary page, review your monitoring configuration for your application group. To create the monitoring template, click Create.  9. You might have to restart IIS or recycle the application pools to finalize the configuration of the applications for monitoring. If a restart or recycle is required, you receive an alert and can use the task link in the knowledge base to perform the necessary action.  Note  After you restart the application it does not begin collecting information accessed by users.  10. If you do want to customize server-side monitoring settings further and to configure and enable client-side monitoring, select the Enable additional configuration options for server-side and client-side monitoring check box, and then click Next. This command adds pages to the wizard as described below. |

Additional Customization for .NET Application Performance Monitoring (Server-Side Perspective)

Using the Modifying Settings page, you can customize server-side monitoring settings for specific application components.

To customize .NET Application Performance Monitoring for a specific application component (server-side perspective)

|  |
| --- |
| 1. If you want to customize server-side monitoring settings further and to configure and enable client-side monitoring, on the Server-Side Configuration page, select the Enable additional configuration options for server-side and client-side monitoring check box, and then click Next. This command adds pages to the wizard.  Warning  If you do not want to change settings of an application component monitor, click Next, and continue with Client-Side Configuration.  2. To select the specific application component for which you want to customize monitoring, on the Server-Side Customization page, click Customize. The Modifying Settings page lets you customize and specialize monitoring for the specific application component and create transactions for individual webpages, web methods, or functions within the application component. When you are finished, click OK, and then click Next. If you do not want to configure and enable client-side monitoring, click Next on the Client-Side Configuration page, and then click Next on the Enable Client-Side Monitoring page.  3. On the Summary page, review your monitoring configuration for your application group. To create the monitoring template, click Create.  4. You might have to restart IIS or recycle the application pools to finalize the configuration of the applications for monitoring. (For System Center 2012 SP1 only: You might have to restart the Windows Service.) If a restart or recycle is required, you receive an alert and can use the task link in the knowledge base to perform the necessary action.  Note  After you restart the application it does not begin to collect information until it is accessed by users. |

Enable and Configure .NET Application Performance Monitoring (Client-Side Perspective)

Client-side application monitoring lets you measure details of the customer experience, such as how long it takes for a page to load. It is another way to monitor how your applications are working from the perspective of your customer. Client-side application monitoring helps you determine whether a problem exists on your server, in the application, or elsewhere.

Important

You can only configure client-side monitoring for applications that have been configured for server-side monitoring.

Important

When working with web applications configured using IIS Shared Configuration, the Operations Manager “Privileged Monitoring Account” Runas Profile associated with the Windows agents hosting the application must have read and write permissions on the shared directory that hosts the web application files to create the Client-Side Monitoring Collector web application, as well as local administrative privileges on each server in the farm to access the IIS metabase for discovery.

Tip

Client-side monitoring can be set up at the same time as server-side monitoring when you run the .NET Application Performance Monitoring wizard or through editing an existing instance of a template as described below.

To enable and configure .NET Application Performance Monitoring (client-side perspective)

|  |
| --- |
| 1. You can either enable client-side monitoring as part of the .NET Application Performance Monitoring Wizard when you configure monitoring for server-side monitoring, or you can revise an existing template to include client-side monitoring. This procedure describes how to enable client-side monitoring while authoring the template.  To enable client-side monitoring, on the Server-Side Configuration page, select the Enable additional configuration options for server-side and client-side monitoring check box and continue with the wizard.  To revise the template to add client-side monitoring, see [To add client-side monitoring to an existing .NET Application Performance Monitoring template](#z43)  2. On the Client-Side Configuration page, you can select to turn on performance and exception event alerts, set page load thresholds and the Ajax and WCF threshold for the application group you are going to monitor. What you enter in the Configure client IP address filter section determines the client requests that are monitored. You can use client IP filters to select the networks that you want to exclude from monitoring. By applying filters, administrators can limit the scope of the monitored computers. By default, the filter is set to localhost, so only connections from browsers started on the local server are instrumented for monitoring. If the IP filter list is empty, all IP addresses are monitored. Any IP addresses that fit the filter definitions are excluded from client-side monitoring. For more information and filtering examples, see [How to Configure IP Address Exclusion Filters for Client-Side Monitoring](#z0529d193725b48bfbfcceb153985e298).  3. To configure more settings for this application group, click Advanced Settings. Here, in addition to settings on the previous page, you can set the sensitivity threshold, that lets you filter out fast-running methods, which reduces overall “noise”, making it easier for you to determine where the problem is and reduces network bandwidth usage. You can also choose to sample only a percentage of the incoming requests. Choosing to monitor only some of the incoming requests can help reduce the load on your monitoring server. Additionally, you can configure these settings:   In the Monitors section, you can change the default thresholds and intervals for the monitors.   In the Data collection section, you can select the type of data you want to collect.  Warning  Enabling the Exception Stack and Global Variables data collection sends application data to the monitored server. We recommend that you do not enable the data collection from Exception Stack and Global Variables unless the application is configured to use an HTTPS protocol.   Load balancer settings let you select the type of load balancers that you are using with your application. You can also add your own load balancer, if it is not included in the list. For more information about load balancers, see [Client-Side Monitoring with Targeted Groups and Load Balancers](#z474bdebf6bbc429fa791a382d95113bc)   In the Monitored Servers section, you can target a group to limit the scope of the monitoring to a group of servers. To select a targeted group, click Search and use the Group Search page to search for the group by name and management pack, and then add them to the selected objects list. The targeted group you select consists of only the servers hosting the web application that set the application pages to return browser-side events. This group lets you limit client-side monitoring independent of server-side monitoring.  Tip  Only applications hosted on servers that are members of both the server-side and client-side targeted groups are monitored by client-side monitoring.  When you have made your changes, click OK, and then click Next.  4. To enable an application group for client-side monitoring, on the Enable Client-Side Monitoring page, select the application you want to enable for client-side monitoring. To customize settings for a selected application component, click Customize.  5. On the Modifying Settings page, you can configure the same settings for the application component that you did for the entire application group with the addition of Excluded pages and Transactions. In the Excluded Pages section, click Add to add the pages that you want to exclude from client-side monitoring. The pages you add to this list are the pages that the Check Client Side Monitoring Compatibility task found incompatible when you ran the task before configuring your application for monitoring. In the Transactions section, click Add to add transactions for ASP.NET webpages.  6. To review all of your monitoring configurations—both server-side configurations and client-side configurations—click the Summary tab. After you have reviewed the configuration, click OK.  Tip  If you want to change any configurations, while you are on this page is a good time to do so. For example, to review or change your server-side configuration, click the Server-Side Configuration page to see that configuration. To disable client-side monitoring, click the Enable Client Side Monitoring page, and clear the check box.  7. On the Summary page, review your monitoring configuration for your application group. To create the monitoring template, click Create.  8. After the client-side monitoring has been configured, you receive an alert to recycle IIS on the affected servers when the client-side monitoring settings have been applied to the server. You can use the link in the knowledge base article to recycle the IIS application pools on the server.  Note  After you restart IIS, an application does not begin to be monitored until it is used. |

To add client-side monitoring to an existing .NET Application Performance Monitoring template

|  |
| --- |
| 1. To enable client-side application monitoring to an existing .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group you configured for server-side monitoring, and then select Properties.  2. On the Properties page, click the Enable Client Side Monitoring tab, and select the Enable check box next to the application group.  Enable client-side monitoring    3. The Customize option on this page opens the Modifying Settings page as described in the previous procedure.  4. To configure client-side default settings, click the Client-Side Defaults tab. These settings and those on the Advanced Settings page are described in the above procedure.  Tip  Only applications hosted on servers that are members of both the server-side and client-side targeted groups are monitored by client-side monitoring.  5. To review all of your monitoring configurations—both server-side configurations and client-side configurations—click the Summary tab. After you have reviewed the configuration, click OK.  6. On the Summary page, review your monitoring configuration for your application group. To create the monitoring template, click Create.  7. After the client-side monitoring has been configured, you receive an alert to recycle IIS on the affected servers when the client-side monitoring settings have been applied to the server. You can use the link in the knowledge base article to recycle the IIS application pools on the server.  Note  After you restart IIS, an application does not begin to be monitored until it is used. |

See Also

[Before You Begin Monitoring .NET Applications](#z43beed71594a40d9aa347210b0e73bcb)

[How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9)

[Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)

How to Add, Enable, and Disable Namespaces

Namespaces define where application performance monitoring should go to collect more detailed information, such as parameters and variables for functions in the specified namespaces or classes. They can be added to the default list and be treated as either entry points or as methods. The default monitoring list can be changed by adding, editing, or removing namespaces and classes.

Note

Many large .NET Framework namespaces (such as System.\* and Microsoft.\*) are disabled by default. To enable them, you need to explicitly add them to the namespaces list as enabled namespaces. Additionally, adding namespaces can affect performance and you should add them only after careful testing.

Note

For System Center 2012 SP1 only: All namespaces is present in the list and enabled for monitoring by default. However, All namespaces, does not include namespaces that have been explicitly disabled by the user or by the default configuration files.

If a namespace is generating too many events, collection for that namespace is turned off by the agent automatically to prevent overloading the agend and an event is written on the agent in the Operations Manager event log. However, everything else will continue to run as expected.

Add a Namespace

To add a namespace

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right-click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Namespaces to open the Namespace list page. This is where you can add namespaces.  5. To add a namespace, on the Namespace list page, select Namespace or class, click Add, and then type the namespace you want to add.  Note  The namespace and class names are case-sensitive. The class name should be specified in the following format: Namespace.ClassName  The Enable monitoring checkbox specifies whether or not you want to collect information from functions contained in the namespace or class. If you want to disable monitoring, perhaps because the functions in the namespace are called very frequently and could be creating performance overhead, clear the Enable monitoring checkbox, and the functions will not be monitored. If you want to make this namespace an entry point, select the Enable as entrypoint checkbox. Performance event timing begins when the monitor encounters a known entry point and completes when the processing of the entry point finishes. If that time exceeds the alerting threshold, then a performance event is reported. For most ASP.NET, ASP.NET MVC, and WCF applications and services, it is typically not necessary to mark specific namespaces or classes as entry points because the default configuration files already contain the entry points that are used by those technologies. Click OK.  Warning  Namespaces are set on the process level. If you enable a namespaces for an application that is running in the process and then disable it for a different application running in that process, there will be a namespace conflict and application monitoring will be disabled. To resolve this, you must make the namespace handling the same for all applications in the same process.  Note  In contrast to namespaces, methods allow you to enable and disable data collection at the specific function level and let you monitor both time measurements as well as collecting deep information, such as variables and parameters used by the methods. |

Edit a Namespace

To edit a namespace

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Namespaces to open the Namespace list page. This is where you can edit namespaces.  5. To edit a namespace, on the Namespace list page, click Edit, make your changes, and then click OK.  Note  The namespace and class names are case-sensitive. The class name should be specified in the following format: Namespace.ClassName |

Remove a Namespace

To remove a namespace

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Namespaces to open the Namespace list page. This is where you can remove namespaces.  5. To remove a namespace, on the Namespace list page, select the namespace or class you want to remove, click Remove, and then click OK. |

When to Enable or Disable Monitoring of All Namespaces and Classes

This setting is used to figure out what you do not know when you are troubleshooting. When you are working with an application, you might not know what you need to enable to get the data you want. This is when you should enable all namespaces. Monitoring all namespaces gives you a view into all the methods that make up the request and ensures that all paths through the application are measured. If, for instance, you only select specific namespaces that you know exist, you might not collect all the information you need to troubleshoot performance problems. You would miss collecting performance information from unlisted namespaces. In this case, the timing in the known functions would include the amount of time spent in the unlisted namespace, giving the false impression that the known functions are slow when they are not.

When to use: Use this when you are trying to troubleshoot a problem and need to isolate the slow function.

Strategy: Before setting specific namespaces, it is good to first set all namespaces to know what’s being hit, and then reduce the number of namespaces to those that are relevant. This reduces amount of data collected and reduces impact on monitored application.

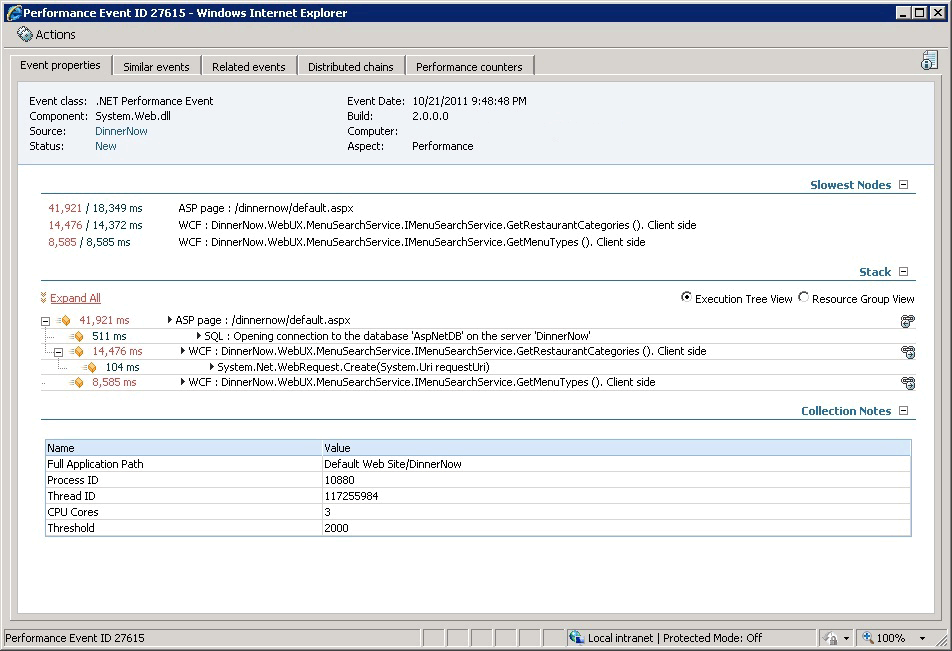
Tip

It is best to enable all namespaces in a test environment before implementing it in a production environment as there is a performance penalty to monitoring all namespaces. When a test environment is not available, begin by measuring a single server’s baseline performance, and then enabling all namespaces on that single server. When you can determine that performance is not impacted, then roll out the setting to more servers.

For System Center 2012 SP1 only: All namespaces is in the list and enabled for monitoring by default. However, All namespaces, does not include namespaces that have been explicitly disabled.

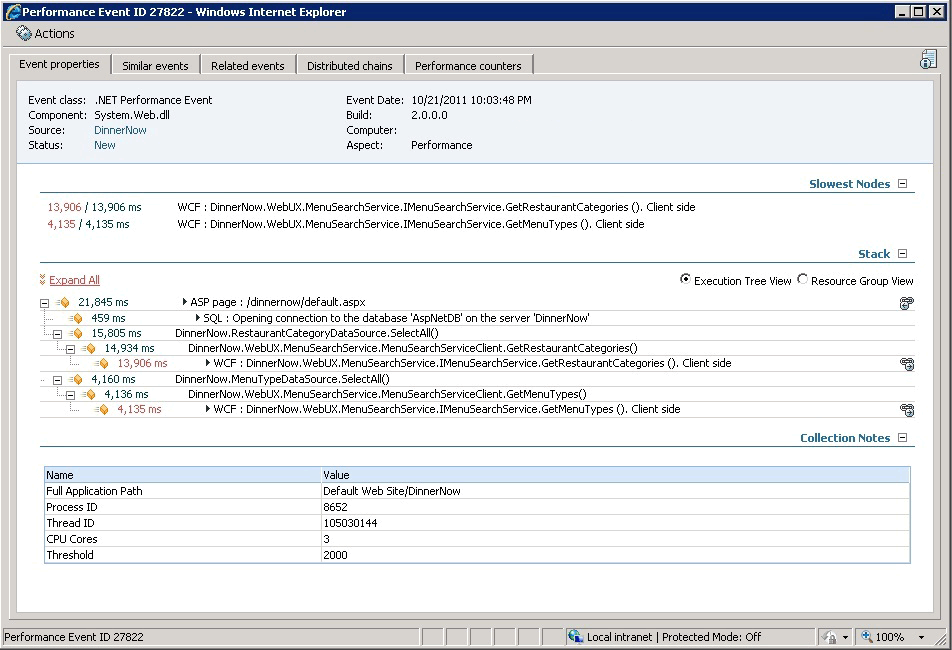
This example shows the difference in monitoring data when you change which namespaces are enabled.

Monitoring with no namespaces



Monitoring with no additional namespaces enabled, the execution tree shows that calls are going to WCF services, but has no details about the actual flow or where those calls are coming from. If a developer tries to investigate this issue and sees that the web service is called from multiple places, the information is incomplete and less useful in helping them find the true root cause.

Monitoring with specific namespaces



By adding the appropriate application specific namespaces, the execution tree shows more details about what is happening in the slow webpage.

To enable or disable monitoring of all namespaces and classes

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Namespaces to open the Namespace list page. This is where you can add namespaces.  5. To add a namespace, on the Namespace list page, select All namespaces, click Add.  6. To enable monitoring for all namespaces, select Enable monitoring. If you want to make all namespaces an entry point, select Enable as entrypoint. Click OK.  Tip  With web applications you typically do not need to enable a namespace or class as an entry point. Entry points are used as the trigger to start measuring performance. .NET provides well-known entry points that should cover the majority of web applications. You need to set the entry point flag only if the monitored application is called through a custom interface. |

See Also

[Understanding and Using Assemblies and Namespaces in .NET](http://go.microsoft.com/fwlink/?LinkId=230965)

How to Add, Edit, and Remove Methods

Methods define entry points used to start measuring for performance events and to monitor for exceptions. If the application defines custom entry points that are not directly called from .NET, these entry points might not be monitored because we don’t that calls to this function are not part of an existing transaction. Methods also let you make fine-grained changes to how data is collected by .NET Application Performance Monitoring. You can create a method for a specific function and then disable monitoring or disable the collection of parameters. This allows you to ensure that functions working with sensitive data are not going to send that data to the development team who might be working with performance events.

Add a Method

To add a method

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Methods to open the Methods list page. This is where you can add methods.  5. To add a method, on the Methods list page, click Add, type the method name and in the Settings section, select if you want to Enable monitoring, set a Sensitivity threshold for this method, and Collect function parameters. Click OK and the application performance monitoring service will add this method to the list of monitored methods.  Important  Adding Methods that are defined in the .NET Framework as part of mscorlib will not produce any effect.  Note  The method name is case-sensitive and should be specified in the following format: Namespace.ClassName.MethodName |

Edit a Method

To edit a method

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Methods to open the Methods list page. This is where you can edit methods.  5. To edit a method, on the Methods list page, click Edit, make your changes, and then click OK.  Note  The method name is case-sensitive and should be specified in the following format: Namespace.ClassName.MethodName |

Remove a Method

To remove a method

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Set Methods to open the Methods list page. This is where you can remove methods.  5. To remove a method, on the Methods list page, select the method you want to remove, click Remove, and then click OK. |

How to Add, Edit, and Remove Exception Tracking

Adding functions to the Exception tracking list allows you to add namespace or classes where Operations Manager tracks exception parameters or variables and collects additional information about each exception that is raised from a namespace or class.

The default list of .NET functions that are monitored for exceptions includes namespaces and functions, some of which have monitoring disabled by default. For more information, see [Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58), which includes a list of default exception handlers.

Warning

Exception tracking is set on the process level. If you add a function to the Exception tracking list for an application that is running in the process and then disable it for a different application running in that process, there will be a conflict and application monitoring will be disabled. To resolve this, you must make the Exception tracking list the same for all applications in the same process.

Add a Class or Namespace to the Exception Tracking List

To add a class or namespace to the exception tracking list

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Exception tracking to open the Exception tracking class or namespace page. This is where you can add classes or namespaces where you intend to collect exceptions.  5. To add a class or namespace, on the Exception tracking class or namespace page, click Add, select whether you want to collect exceptions for a particular Namespace or class or for All namespaces, and then type the class or namespace you want to add. The Enable monitoring checkbox is selected by default, which means that you want to enable monitoring for this class or namespace. To explicitly disable monitoring of exceptions in a particular class or namespace, clear the checkbox. Click OK.  For System Center 2012 SP1 only: All namespaces is present and enabled in this list by default. However, All namespaces, does not include namespaces that have been explicitly disabled.  Important  Adding Exception Tracking for namespaces or classes that are defined in the .NET Framework as part of mscorlib will not produce any effect.  Note  The Namespace and Class names are case-sensitive and should be specified in the following format: Namespace.ClassName |

Edit a Class or namespace on the Exception Tracking List

To edit a class or namespace on the exception tracking list

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Exception tracking to open the Exception tracking class or namespace page. This is where you can edit the list of classes or namespaces where to look for exceptions.  5. To edit a class or namespace, on the Exception tracking class or namespace page, click Edit, modify the settings you want to change, and then click OK.  Note  The Namespace and Class names are case-sensitive and should be specified in the following format: Namespace.ClassName |

Remove a Class or Namespace on the Exception Tracking List

To remove a class or Namespace on the exception tracking list

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group whose settings you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Exception tracking to open the Exception tracking class or namespace page. This is where you can remove classes or namespaces from the list.  5. To remove a class or namespace, on the Exception tracking class or namespace page, click Edit, modify the settings you want to change, and then click OK.  Note  The Namespace and Class names are case-sensitive and should be specified in the following format: Namespace.ClassName |

Using Exception Handlers to Define Critical Exceptions

Exception handlers are application functions that “catch” exceptions that the applications throw to report errors and perform some error handling. By default, .NET Application Performance Monitoring defines critical exceptions as exceptions handled by specific exception handlers provided by the .NET framework. These handlers catch top-level ASP.NET exceptions, and web service exceptions that the monitored application failed to catch and handle internally. By adding exception handlers, you are adding to what application monitoring’s definition of what a critical exception is. In effect, any exceptions handled by these functions will be considered critical exceptions. The advantage to doing this is that you maintain the benefit of streamlined reporting of critical exceptions only, but you have the additional benefit of reporting functions that are of interest to you. It is common to add any customer error handlers defined for web applications to the list of critical exception handlers so that you can be alerted when a user is sent to your error handler page in the web application.

Warning

Exception handlers are set on the process level. If you enable an exception handler for an application that is running in the process and then disable it for a different application running in that process, there will be a configuration conflict and application monitoring will be disabled. To resolve this, you must make the exception handling the same for all applications in the same process.

Default Exception Handlers

The default list of exception handlers includes:

 System.Web.HttpApplication.RecordError

 System.Web.UI.Page.HandleError

 System.Web.Services.Protocols.WebServiceHandler.WriteException

 System.AppDomain.OnUnhandledExceptionEvent

 System.Windows.Forms.Application.ThreadContext.OnThreadException

 System.AppDomain.OnUnhandledExceptionEvent

 System.Runtime.Remoting.Messaging.ReturnMessage..ctor

 System.Windows.Forms.DataGridView.OnDataError

For System Center 2012 SP1 these resources are included:

 Microsoft.Office.Server.Data.SqlSession.LogException

 Microsoft.Office.Excel.Server.CalculationServer.Proxy.ExcelServerProxy.ProcessSoapException

 Microsoft.Office.Excel.Server.CalculationServer.Proxy.ExcelServerProxy.ProcessWebException

 Microsoft.SharePoint.Portal.WebControls.BusinessDataWebPart.ConstructErrorMessage

 Microsoft.SharePoint.Diagnostics.ULS.SendEventTag

 Microsoft.SharePoint.ApplicationRuntime.SPRequestModule.IsWebPartOnExceptionStack

 Microsoft.SharePoint.Utilities.SqlSession.LogException

 Microsoft.Office.Web.Environment.Sharepoint.Diagnostics.ULS.SendExceptionTag

 Microsoft.SharePoint.Diagnostics.ULS.SendExceptionTag

 Microsoft.Office.Server.Diagnostics.ULS.SendExceptionTag

 System.Workflow.Runtime.WorkflowExecutor.IsIrrecoverableException

 System.ServiceModel.DiagnosticUtility.IsFatal

 System.Web.Mvc.ControllerActionInvoker.InvokeExceptionFilters

Add an Exception Handler

To add an exception handler

|  |
| --- |
| 1. To open the .NET Application Performance Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group you want to modify, and then select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Critical Exceptions to open the Exception handlers list page. This is where you can add exception handlers.  5. To add an exception handler, click Add and type the method you want to add to the exception handlers list. If you want this exception handler to affect monitoring, make sure the Enable monitoring checkbox is selected. Click OK.  Important  Adding handlers that are defined in the .NET Framework as part of mscorlib as Critical Exceptions will not produce any effect.  Note  The method name is case sensitive and should be specified in the following format: Namespace.ClassName.MethodName |

Edit an Exception Handler

To edit an exception handler

|  |
| --- |
| 1. Open the .NET Application Performance Monitoring template. In the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group you want to modify and select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Critical Exceptions. This opens the Exception handlers list page where you can edit exception handlers.  5. To edit an exception handler, click Edit, select the exception handler you want to change, and then modify it. Click OK.  Note  The method name is case sensitive. Additionally, the method name should be specified in the following format: Namespace.ClassName.MethodName |

Remove an Exception Handler

To remove an exception handler

|  |
| --- |
| 1. Open the .NET Application Performance Monitoring template. In the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, and then click .NET Application Performance Monitoring.  2. Right click the application group you want to modify and select Properties.  3. On the Server-Side Defaults tab, click Advanced Settings.  4. On the Advanced settings page, click Critical Exceptions. This opens the Exception handlers list page where you can remove exception handlers.  5. To remove an exception handler, select the exception handler you want to remove, click Remove, and then click OK. |

Client-Side Monitoring with Targeted Groups and Load Balancers

When load balancers are used during client-side monitoring, the load balancer is designed to get the true client IP addresses. In a simple case, when a client is connected directly to one web server, the web server knows the client IP address. However, when you have several servers and use a load balancer to distribute traffic from the clients among the servers, this can present two problems. First, the IP reported to the collector is the virtual IP that the load balancer uses, not the real end-user IP. When users hit the load balancer, it sends them to an appropriate web server. Due to the load balancer, the web server sees only the internal port IP (inside the network), not the real IP (outside the network). Additionally, if some servers are monitored and some are not, data can be lost because responses from the clients are often sent to servers in a least load or round robin fashion, which means that the response might go to a server that is not hosting a collector.

Here are some strategies for setting up client-side monitoring when you have multiple monitored servers and use load balancers.

Monitoring with a Load Balancer and Targeted Group

When you configure client-side monitoring, you have the option to set the target group, limiting the number of web servers used for monitoring. In this scenario, only monitored servers in the target group will inject the JavaScript used for monitoring and the servers outside the target group that are not monitored will not get instrumented when using load balancers with the web servers. This results in having incorrect data. The load balancer does not know which servers are inside or outside the targeted group and sends client requests to servers that are both inside (monitored) or outside (unmonitored) of the targeted group. The result is that requests that have been instrumented and try to return data to the collectors might send their results to servers that cannot handle the data.

 Solution 1 If you are authoring a new .NET Application Performance Monitoring template and including client-side monitoring for a targeted group, we recommend that you choose a group of servers that are all served by the same load balancer. Target monitoring to all servers in the load-balanced farm.

 Solution 2 If you are already running client-side monitoring with a targeted group and a load balancer, you can resolve this issue by creating a rule on the load balancer that directs all of the monitoring traffic to the monitored servers. Monitoring traffic is the monitoring JavaScripts that send data to the collector endpoint. You can identify monitoring traffic because it contains /CSMCollector in the URL. Each load balancer has its own model for configuring rules. Refer to your load balancer’s documentation for details about how to create the rule.

How to Configure IP Address Exclusion Filters for Client-Side Monitoring

You can use client IP filters to choose the networks that you want to monitor. By applying filters, administrators can limit the scope of the monitored computers. By default, only localhost is monitored. If the IP filter list is empty, all IP addresses are monitored. Any IP addresses that fit the filter definitions are excluded from client-side monitoring.

The following examples show how to use IP filters to monitor IP addresses.

Exclude a Set of IP Addresses from Monitoring

Example: To monitor all IP addresses except 192.168.\*.\*

|  |
| --- |
| 1. To monitor all IP addresses, except some (IP addresses 192.168.\*.\* in this example), on the Client Side Configuration tab, in the Configure client IP filter section, click Add.  2. On the Client IP address filter page, set the Comparison Type to IP is in subnet.  Note  Comparison type has two values: In subnet excludes user IP addresses that match the IP addresses in the subnet. Not in subnet excludes the user IP addresses that do not match the IP addresses in the subnet.  3. In the IP address box, enter the IP addresses that you want to exclude from monitoring. In this example, enter 192.168.0.0.  4. In the Netmask box, enter the part of the filter IP address and user IP address that have to be compared for equality. In this example, enter 255.255.0.0.  5. Click OK. |

Monitor a Specific Set of IP Addresses

Example: To monitor the specific set IP addresses 192.168.\*.\*

|  |
| --- |
| 1. To monitor a specific set of IP addresses (in this example, IP addresses 192.168.\*.\*), on the Client Side Configuration tab, in the Configure client IP filter section, click Add.  2. On the Client IP address filter page, set the Comparison Type to IP is not in subnet.  3. In the IP address box, enter the IP addresses that you want to monitor. In this example, enter 192.168.0.0.  4. In the Netmask box, enter the part of the filter IP address and user IP address that have to be compared for equality. In this example, enter 255.255.0.0.  5. Click OK. |

Monitor a Specific Set of IP Addresses and Exclude Some IP Addresses from that Set

Example: To monitor IP addresses 192.168.\*.\* and exclude IP addresses 192.168.10.\*

|  |
| --- |
| 1. To monitor a specific set of IP addresses and exclude some from that set (in this example, only IP addresses 192.168.\*.\*, except IP addresses 192.168.10.\*), you have to configure two filters. First, create the filter to scope to only IP addresses 192.168.\*.\*, and then a filter that excludes 192.168.10.\*. To create the first filter to monitor only IP addresses 192.168.\*.\*, on the Client Side Configuration tab, in the Configure client IP filter section, click Add.  2. Set the Compare Type to IP is not in subnet.  3. In the IP address box, enter the IP addresses that you want to monitor. In this example, enter 192.168.0.0.  4. In the Netmask box, enter the part of the filter IP address and user IP address that have to be compared for equality. In this example, enter 255.255.0.0.  5. Click OK.  6. Then, create the second IP filter to exclude IP addresses 192.168.10.\*. On the Client Side Configuration tab, in the Configure client IP filter section, click Add.  7. Set the Compare Type to IP is in subnet.  8. In the IP address box, enter the IP addresses that you want to exclude from monitoring. In this example, enter 192.168.10.0.  9. In the Netmask box, enter the part of the filter IP address and user IP address that have to be compared for equality. In this example, enter 255.255.255.0.  10. Click OK. |

When to Use the IPv4 and IPv6 Filters

If the IPv4 protocol is the only protocol that is enabled on the web server, leave the Use IPv6 check box blank.

If the IPv6 protocol is enabled on the web server, select the Use IPv6 check box to add the IPv6 filter.

How to Start Monitoring a New Application

When you have a new application that you are learning about, you can use the .NET Application Performance Monitoring template in System Center 2012 – Operations Manager to configure monitoring for it. Here are some settings to start with that will help you get to know your new application. In addition, it is ideal that you begin monitoring in a test or development environment.

Monitoring Settings for a New Application

Following this strategy for monitoring a new application will help you get to know how the application behaves within your system and for your customer.

Start Monitoring Server-Side Only with a Simple Monitored System and Short-Term Settings

First, keep the configuration simple: monitor one application on one server. Second, when you first configure .NET Application Performance Monitoring to monitor a new application, plan to keep the settings you implement long enough for you to understand some trends. A day’s worth of data should provide you with insight into the performance and usage patterns of the application.

Establish Baseline Performance Using Default Settings and Some Specific Settings

For the most part, you will want to keep default settings. The default settings ensure that you will see any large issues with the application and keep the impact on the monitored application at a minimum.

If you are not getting any performance or exception events raised, you can use the following steps to get a feel for what the baseline performance looks like.

To begin monitoring, here are some settings you might want to adjust as noted here:

 Lower the thresholds for performance. This will help you establish a baseline performance measure by seeing what the current performance characteristics of the application are.

 Enable all namespaces. You want to find out what namespaces are involved and if you set specific namespaces at first, you might miss a namespace where an error is occurring.

 Collect all exceptions, not just critical exceptions. You need to know what kinds of exceptions are being thrown. Using known exception handlers limits the exceptions you will receive.

This can result in a lot of data—more than you would want for long-term monitoring—but at first, this amount of data will be helpful as you will see trends, such as the kinds of paths customers are taking through the system and what normal performance looks like.

With the data collection complete, use the Application Advisor reports, such as Application Performance Analysis, to see how the monitored applications are looking. Using the report you will see what the average duration is for the heaviest (longest running) calls through the system as well as the maximum amount of time spent processing requests. This allows you to set customized smart thresholds based on real application performance. You will also see which functions are running faster than others, and you can create specific web page, web method, and function transactions for the critical methods so that you can ensure they are responding under a tighter SLA than the application as a whole. For more information on viewing reports, see how to scope and run and Application Advisor report in Prioritizing Alerts by Using Application Advisor.

Adjust Settings and Compare to the Baseline

Once you have established a baseline performance measure, begin to adjust the settings to tune the monitoring so it catches the kinds of exceptions that are being raised. By reporting all exceptions, you will see if there are any default exception handlers in the application that are catching exceptions for which you would prefer receiving alerts. The data you get will be more meaningful and lower in volume with each adjustment.

 Remove the custom settings and set thresholds based on the data collected.

 Add specific namespaces based on the call stacks in the performance and exception events you found during the baseline phase.

 Add exception handlers for any application level “catch all” handlers that keep exceptions from going outside the application and to the .NET Framework exception handlers.

 Add specialized transactions to monitor the performance of common methods that should be held to a stronger SLA than the application as a whole.

Compare the new data to your baseline. You will begin to see the real average response time, for instance. Now that you know the various performance exceptions the application is sending, you can add the specific namespaces you want rather than monitoring all namespaces. Your application will be configured to be monitored based on the observed performance levels and will be alerted if things move outside of normal levels.

Gradually Deploy the Application to More Monitored Servers in Your System

After monitoring the application for a time with the new monitoring configuration, when you feel your application is healthy, increase the number of servers you are running the application on and monitoring from one to 10, for example. Once you have it running healthy at that level, increase the deployment and monitoring to more servers, and so on. This gradual rollout approach will help you gain confidence in the monitoring for that application and help ensure the health of your system.

Begin Client-Side Monitoring

When you are confident that your application is running well within your system, it’s a good time to monitor what the customer experiences. This is what client-side application monitoring does. To enable client-side monitoring, see [How to Configure Monitoring for .NET Applications](#z18139deace8442b8bd50d467d72623e1)

What the Operator can do with This Information

Using this basic information, the operator can have a better idea where the problem is with the application or with the infrastructure and know whether it is something only to the development team can fix or the operator can address directly.

See Also

[How to Add, Enable, and Disable Namespaces](#z525024710857408fb17e73733225f65b)

[Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58)

[Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)

[Application Monitoring Using the Default Settings](#zabd195c1584a4eac9c4554ede7d36845)

Authoring Strategies for .NET Application Monitoring

Here are some scenarios and settings to change during authoring that can help you receive the monitoring experience and data that are most helpful for you.

Monitoring a New Application for which the Administrator has Little Knowledge

Accepting all defaults can be a good way to start monitoring an application for which the administrator has very little or no knowledge. Then, after monitoring with all defaults for some time, the administrator can begin adjusting settings based on the monitoring alerts, Application Diagnostics data, and Application Advisor reports. For more information, see [How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9) and [Application Monitoring Using the Default Settings](#zabd195c1584a4eac9c4554ede7d36845)

Limit Monitoring to a Specific Set of Servers

Defining a targeted group allows you to limit monitoring to a specific set of servers. In the .NET Application Performance Monitoring wizard, targeted group for server-side monitoring is on the What to Monitor page. Targeted group for client-side monitoring is on the Enable Client-Side Monitoring page. If you are using a targeted group for client-side monitoring and use a load balancer, see [Client-Side Monitoring with Targeted Groups and Load Balancers](#z474bdebf6bbc429fa791a382d95113bc)

For very large application deployments, you typically do not need to monitor all instances of the application. A representative sample is enough to get the data you need. Using only a representative sample will keep the amount of data collected and stored lower.

Reduce the “Noise” by Defining How Much Data You Collect

Increasing the sensitivity threshold allows you to filter out fast-running methods, which reduces overall “noise”, or how deep the call stack is going to go, making it easier for you to determine where the problem is. It also reduces network bandwidth usage.

The sensitivity setting is used to determine if a function call should be included in the call stack. Any function that executes and returns faster than the sensitivity level is dropped, keeping small fast-running functions from hiding the actual problem. Remember that using sensitivity only reduces the number of functions shown in the call stack for specific events, but an event will still be generated if the overall threshold is surpassed.

You can adjust the sensitivity threshold for server-side and client-side monitoring independently.

To change the sensitivity threshold for server-side monitoring

|  |
| --- |
| 1. To open properties for the application group that you want to reconfigure, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group that you want to want to configure, and then select Properties.  Note  If you are currently authoring a new .NET Application Performance Monitoring template, to change the sensitivity threshold for server-side monitoring, go to the Server-Side Configuration page and click Advanced Settings Change the Sensitivity threshold and click OK.  2. To change the sensitivity threshold for server-side monitoring, on the Properties page, click the Server-Side Monitoring tab, and then click the Advanced Settings button.  3. Change the Sensitivity threshold and click OK. |

To change the sensitivity threshold for client-side monitoring

|  |
| --- |
| 1. To open properties for the application group that you want to reconfigure, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group that you want to want to configure, and then select Properties.  Note  If you are currently authoring a new .NET Application Performance Monitoring template, to change the sensitivity threshold for client-side monitoring, go to the Client-Side Configuration page and click Advanced Settings. Change the Sensitivity threshold and click OK.  2. To change the sensitivity threshold for client-side monitoring, on the Properties page, click the Client-Side Monitoring tab, and then click the Advanced Settings button.  3. Change the Sensitivity threshold and click OK. |

It is also possible for high sensitivity to hide problems. In the situation where you have a function that calls another function, if the callee’s response time increases even slightly, it might cause issues for the application. For example, if you have a data processing function that calls a lookup function 1,000 times and the lookup’s processing time increases by 1 ms, you will increase the response time for your top level function by a full second. This might be masked by the high sensitivity. When you find this kind of situation, you can add the callee as a method and set a custom sensitivity for it to ensure it is always measured according to the lower sensitivity threshold.

Application failure alerts are application, or code, failures that are detected within the application. You can choose not to receive application failure alerts, which will potentially occur very often if an application has problems because these kinds of alerts usually require code modifications to address. Turning this off reduces the “noise” of many alerts raised that cannot be directly resolved by the operations team.

You can turn off application failure alerts for server-side and client-side monitoring independently.

To turn off alerts for application failures for server-side monitoring

|  |
| --- |
| 1. To open properties for the application group that you want to reconfigure, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group that you want to want to configure, and then select Properties.  Note  If you are currently authoring a new .NET Application Performance Monitoring template, to turn off alerts for application failures for server-side monitoring, go to the Server-Side Configuration page and click Advanced Settings. Clear the Application failure alerts checkbox and click OK.  2. To turn off application failure alerts for server-side monitoring, on the Properties page, click the Server-Side Defaults tab, and then click the Advanced Settings button.  3. On the Advanced settings page and clear the Application failure alerts checkbox.  4. Click OK. |

To turn off alerts for application failures for client-side monitoring

|  |
| --- |
| 1. To open properties for the application group that you want to reconfigure, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group that you want to want to configure, and then select Properties.  Note  If you are currently authoring a new .NET Application Performance Monitoring template, to turn off alerts for application failures for client-side monitoring, go to the Client-Side Configuration page and click Customize. On the Modifying Settings page, in the Transactions section, click Add. On the Transaction Properties page, clear the Application failure checkbox and click OK.  2. To turn off application failure alerts for client-side monitoring, on the Properties page, click the Client-Side Monitoring tab, and then click the Advanced Settings button.  3. In the Transactions section, click Add.  4. On the Transaction Properties page, clear the Application failure checkbox.  5. Click OK. |

Only Receive Critical Exceptions

By default, .NET Application Performance Monitoring defines critical exceptions as exceptions handled by specific exception handlers provided by the .NET framework. These handlers catch top-level ASP.NET exceptions and web service exceptions that the monitored application failed to catch and handle internally. By adding exception handlers, you are adding to what application monitoring’s definition of what a critical exception is. In effect, any exceptions handled by these functions will be considered critical exceptions. The advantage to using exception handlers is that you maintain the benefit of streamlined reporting of critical exceptions only, but you have the additional benefit of reporting functions that are of interest to you. For more information and a list of default exception handlers, see [Using Exception Handlers to Define Critical Exceptions](#z67a9a184f90c4745a3eca66c9d006c58).

Improve Client-Side Monitoring Performance and Reduce Load on Your Server

You might also want to adjust the sampling rate to control the performance impact of the monitoring on your application with client-side monitoring. Reducing the sampling rate reduces the application monitoring traffic and helps conserve server resources. If you have even a low-traffic site, instrumenting and collecting data from every user who connects will result in a large amount of non-actionable data to sift through. Taking a random sample will give you the insight you need into the application performance from the client perspective without flooding you with a large amount of data to process and store.

To change the sampling rate for client-side monitoring

|  |
| --- |
| 1. To open client-side properties for the application group that you want to reconfigure, in the Operations Manager console, in the navigation pane, click the Authoring button, expand Management Pack Templates, click .NET Application Performance Monitoring, right-click the application group that you want to want to reconfigure, and then select Properties.  On the Properties page, click the Client-Side Defaults tab, and then click the Advanced Settings button.  Note  Because you can change the sampling rate for both the application group and each application component, changes to the application group settings will not automatically be applied to the component settings when the component settings have been previously customized.  2. In the Sampling section, use the drop-down menu to select the percentage of incoming requests that you want to monitor. For example, if you select 50%, you will monitor 50 percent of the incoming requests. Select 25% and you will monitor 25 percent of the incoming requests, and so on. To get helpful information, you do not have to monitor all of the incoming requests.  3. When you have set the sampling rate, click OK. |

See Also

[How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9)

Application Monitoring Using the Default Settings

Accepting all defaults can be a good way to start monitoring an application for which the administrator has very little or no knowledge. Then, after monitoring with all defaults for some time, the administrator can begin adjusting settings based on the monitoring alerts, Application Diagnostics data, and Application Advisor reports.

Using Default Settings for Server-Side Monitoring

You still need to select the application you want to monitor and the target management pack, but then you can start monitoring with “all defaults”. With all default settings Application Performance Monitoring will monitor only server-side, and all thresholds for all pages will be the same. To see the default values, you can go through the wizard without changing anything.

Using Default Settings for Client-Side Monitoring

The defaults are enough to get this started and to allow you to test it out from localhost connections. It is simply scoped to monitor localhost by default.

You can certainly accept the default settings for client-side monitoring, but it is very important to run the compatibility check task to validate if the application can be monitored and if any of the pages should be excluded from monitoring. Therefore, simply applying of client-side monitoring defaults might be risky. For more information about running the compatibility check task, see [Before You Begin Monitoring .NET Applications](#z43beed71594a40d9aa347210b0e73bcb)

In general, client-side threshold settings should be higher than server-side threshold settings. This is because the client-side monitoring contains the server time, too. For instance, when a client-side event is divided into various parts, some of the time is spent on the server, but the client also monitors the time spent on the network and the time spent in the browser.

IP Address Filters and Load balancers Also, IP filters by default will enable client-side monitoring for localhost only. Additionally, load balancer settings need to be set correctly for client-side monitoring. If the IP header is not set, all client-side monitoring traffic will appear to be coming from a single IP address. All IP- and subnet- based reports in Application Advisor will be invalid if the default settings are used with a load balancer.

There is no default for the load balancers. The load balancer setting is one you can opt to change, whereas you must change the client IP filters because if you do not update those settings you will not get any data at all.

IP Address Filters You can use client IP filters to choose the networks that you want to monitor. By applying filters, administrators can limit the scope of the monitored computers. By default, only localhost IP addresses are monitored. If the IP filter list is empty, all IP addresses are monitored. Any IP addresses that fit the filter definitions are excluded from client-side monitoring. For more information, see [How to Configure IP Address Exclusion Filters for Client-Side Monitoring](#z0529d193725b48bfbfcceb153985e298)

See Also

[Authoring Strategies for .NET Application Monitoring](#z23708fb345b04e4180dd83522eb47830)

[How to Start Monitoring a New Application](#zcc1f2e640fa04d7f81271e4586b9e2d9)

Monitoring SharePoint 2010 Applications in System Center 2012 SP1

The information in this topic applies only to System Center 2012 SP1.

SharePoint application technologies provide various extensibility and customization mechanisms. Custom application code running in SharePoint environment can extend or replace standard pages, can implement custom business rules, often integrates with third-party components, and custom solutions written to operate using SharePoint framework. The customizations and code errors in custom components can affect server performance, significantly impacting overall application experience.

Operations Manager lets you monitor SharePoint 2010 web front-end components. You can monitor standard and custom SharePoint webpages for performance degradation and server-side exceptions.

Warning

Client-side .NET Application Performance Monitoring (APM) is not supported for SharePoint. Enabling client-side .NET Application Performance Monitoring for SharePoint can result in unpredictable application behavior and failures.

Monitoring for SharePoint Applications

SharePoint web front-end components are natively discovered by the IIS management pack as endpoints. You can enable monitoring for SharePoint applications in much the same way you enable monitoring for other .NET web applications. Use the .NET Application Performance Monitoring template to configure SharePoint application monitoring. For more information, see [.NET Application Performance Monitoring Template](#z2166e40acc7d4534ad8e73a51ac2bf99).

When monitoring SharePoint applications for performance violations, events in Application Diagnostics contain additional information, such as SharePoint server API calls and Web Part calls. For each API call, the APM agent will collect relevant SharePoint methods parameters. The APM agent also tracks the execution time for the slowest calls. When detecting performance issues from standard SharePoint webpages using web parts, the slowest resource calls are shown and poorly performing web part names are collected next to the location of the web part on the SharePoint page. When detecting performance issues from custom SharePoint webpages, only the SharePoint API calls are shown.

When monitoring SharePoint applications for exceptions, the exception call stack contains the relevant SharePoint specific parameters available for troubleshooting.

Appendix for .NET Application Monitoring

Information in this appendix is the default entry points and default methods, which includes resource calls and functions that are included with .NET Application Performance Monitoring in System Center 2012 – Operations Manager.

See Also

[Default Entry Points for .NET Application Monitoring](#z2186d3b4528b485cabec8ee4e9dfb8c6)

[Default Methods for .NET Application Monitoring](#z6874fc3d646b46e7965c9ec6bac88500)

Default Entry Points for .NET Application Monitoring

Application Performance Monitoring in Operations Manager is preconfigured with many well-known entry points (see below), but also lets you extend the default list by defining your own entry points. In addition to adding functions as entry points, you may also define entire Namespaces to act as entry points, so that the system begins timing execution the first time that it encounters the namespace during execution. Additionally, the application monitoring agent collects the values of variables for each entry point at the time the event occurs.

Entry points Monitored by Default

For ASP.NET pages

 System.Web.UI.Page.ProcessRequest

 (System.Web.IHttpHandler).ProcessRequest

For ASP.NET 2.0 asynchronous pages

 For System Center 2012 SP1 only: System.Web.UI.Page.AsyncPageBeginProcessRequest

 System.Web.Services.Protocols.LogicalMethodInfo.Invoke



For COM+ server-side

 System.EnterpriseServices.ServicedComponent.RemoteDispatchHelper

For .NET Remoting server-side

 System.Runtime.Remoting.Messaging.ServerObjectTerminatorSink.AsyncProcessMessage

 System.Runtime.Remoting.Messaging.ServerObjectTerminatorSink.SyncProcessMessage

For WCF Server Side

 System.ServiceModel.Dispatcher.SyncMethodInvoker.Invoke

For WWF

 System.Workflow.Runtime.WorkflowExecutor.RunScheduler

For AJAX.NET

 System.Web.Script.Services.RestHandler.ExecuteWebServiceCall

For Windows Services

For System Center 2012 SP1 these entry points are included:

 System.ServiceProcess.ServiceBase.ServiceCommandCallback

 System.ServiceProcess.ServiceBase.ServiceQueuedMainCallback

 System.ServiceProcess.ServiceBase.Stop

 System.ServiceProcess.ServiceBase.DeferredPowerEvent

 System.ServiceProcess.ServiceBase.DeferredSessionChange

For MVC

For System Center 2012 SP1 these entry points are included:

 System.Web.Mvc.ControllerBase.Execute

 (System.Web.Mvc.IController).Execute

 System.Web.Mvc.AsyncController.BeginExecute

 (System.Web.Mvc.IAsyncController).BeginExecute

 System.Web.Mvc.ViewResultBase.ExecuteResult

Default Methods for .NET Application Monitoring

Application Performance Monitoring in Operations Manager includes many functions and resources calls that are monitored by default.

Functions Monitored by Default

Application Performance Monitoring includes many well-known Microsoft .NET Framework functions that are monitored for slow performance or exception data collection.

For SharePoint

For System Center 2012 SP1 these resources are included:

 Microsoft.Office.Excel.WebUI.ExcelWebRenderer.OnPreRender

 Microsoft.Office.Excel.WebUI.InternalEwr.OnPreRender

 Microsoft.SharePoint.Portal.WebControls.ContactFieldControl.OnPreRender

 Microsoft.SharePoint.Portal.WebControls.IndicatorWebpart.OnPreRender

 Microsoft.SharePoint.Portal.WebControls.KPIListWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.ContentEditorWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.ContentEditorWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.ListFormWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.ListFormWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.ListViewWebPart.CreateChildControls

 Microsoft.SharePoint.WebPartPages.ListViewWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.ListViewWebPart.OnInit

 Microsoft.SharePoint.WebPartPages.ListViewWebPart.OnPreRender

 Microsoft.SharePoint.Portal.WebControls.BusinessDataActionsWebPart.RenderWebPart

 Microsoft.SharePoint.Portal.WebControls.GroupDetailWebPart.RenderWebPart

 Microsoft.SharePoint.Portal.WebControls.RelatedGroupsWebPart.RenderWebPart

 Microsoft.SharePoint.Portal.WebControls.WebPartLoc.OnPreRender

 Microsoft.SharePoint.Portal.WebControls.WebPartLoc.RenderWebPart

 Microsoft.SharePoint.Search.Internal.WebControls.SearchPagingWebPart.OnPreRender

 Microsoft.SharePoint.Search.Internal.WebControls.SearchPagingWebPart.RenderWebPart

 Microsoft.SharePoint.Search.Internal.WebControls.SearchStatsWebPart.OnPreRender

 Microsoft.SharePoint.Search.Internal.WebControls.SearchStatsWebPart.RenderWebPart

 Microsoft.SharePoint.Search.Internal.WebControls.SearchSummaryWebPart.RenderWebPart

 Microsoft.SharePoint.Search.Internal.WebControls.SearchSummaryWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.ImageWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.PageViewerWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.XmlWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.XmlWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.AggregationWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.ChartViewWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.DataViewWebPart.OnPreRender

 Microsoft.SharePoint.WebPartPages.DataViewWebPart.RenderWebPart

 Microsoft.SharePoint.WebPartPages.ListFormWebPart.OnLoad

 Microsoft.SharePoint.WebPartPages.ListFormWebPart.OnInit

 Microsoft.SharePoint.Portal.WebControls.BusinessDataWebPart.GetCallbackResult

 Microsoft.SharePoint.Portal.WebControls.BusinessDataDetailsWebPart.CreateChildControls

 Microsoft.SharePoint.Portal.WebControls.BusinessDataListWebPart.CreateChildControls

 Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Entity.ExecuteInternal

 Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.WebService.WebServiceSystemUtility.ExecuteStatic

 Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbSystemUtility.ExecuteStatic

 Microsoft.SharePoint.WebPartPages.DataFormWebPart.OnPreRender

For SharePoint: Base Classes Monitoring

For System Center 2012 SP1 these resources are included:

 System.Web.UI.WebControls.WebParts.WebPart.OnDataBinding

 System.Web.UI.WebControls.WebParts.WebPart.OnPreRender

 System.Web.UI.WebControls.WebParts.WebPart.OnInit

 System.Web.UI.WebControls.WebParts.WebPart.OnLoad

 System.Web.UI.WebControls.WebParts.WebPart.Render

 System.Web.UI.WebControls.WebParts.WebPart.RenderWebPart

 System.Web.UI.WebControls.WebParts.WebPart.OnUnload

 [System.Web.UI.WebControls.WebParts.WebPart].OnDataBinding

 [System.Web.UI.WebControls.WebParts.WebPart].OnPreRender

 [System.Web.UI.WebControls.WebParts.WebPart].OnInit

 [System.Web.UI.WebControls.WebParts.WebPart].OnLoad

 [System.Web.UI.WebControls.WebParts.WebPart].Render

 [System.Web.UI.WebControls.WebParts.WebPart].RenderWebPart

 [System.Web.UI.WebControls.WebParts.WebPart].OnUnload

 [Microsoft.SharePoint.WebPartPages.WebPart].OnDataBinding

 [Microsoft.SharePoint.WebPartPages.WebPart].OnPreRender

 [Microsoft.SharePoint.WebPartPages.WebPart].OnInit

 [Microsoft.SharePoint.WebPartPages.WebPart].OnLoad

 [Microsoft.SharePoint.WebPartPages.WebPart].Render

 [Microsoft.SharePoint.WebPartPages.WebPart].RenderWebPart

 [Microsoft.SharePoint.WebPartPages.WebPart].OnUnload

For SharePoint: SPrequest Methods, Basic SharePoint 2007 APIs

For System Center 2012 SP1 these resources are included:

 Microsoft.SharePoint.Library.SPRequest.AccessContentTypes

 Microsoft.SharePoint.Library.SPRequest.AddField

 Microsoft.SharePoint.Library.SPRequest.AddGroup

 Microsoft.SharePoint.Library.SPRequest.AddMeeting

 Microsoft.SharePoint.Library.SPRequest.AddMeetingFromEvent

 Microsoft.SharePoint.Library.SPRequest.AddMeetingFromICal

 Microsoft.SharePoint.Library.SPRequest.AddNavigationNode

 Microsoft.SharePoint.Library.SPRequest.AddOnProvision

 Microsoft.SharePoint.Library.SPRequest.AddOrDeleteUrl

 Microsoft.SharePoint.Library.SPRequest.AddOrUpdateItem

 Microsoft.SharePoint.Library.SPRequest.AddRoleDef

 Microsoft.SharePoint.Library.SPRequest.AddSubscription

 Microsoft.SharePoint.Library.SPRequest.AddWebPart

 Microsoft.SharePoint.Library.SPRequest.AddWebPartPageToLibrary

 Microsoft.SharePoint.Library.SPRequest.AddWebPartRightsCheck

 Microsoft.SharePoint.Library.SPRequest.AddWorkflowAssociation

 Microsoft.SharePoint.Library.SPRequest.AddWorkflowToListItem

 Microsoft.SharePoint.Library.SPRequest.AddWorkItem

 Microsoft.SharePoint.Library.SPRequest.ApplyAutoHyperLinking

 Microsoft.SharePoint.Library.SPRequest.ApplyTheme

 Microsoft.SharePoint.Library.SPRequest.ApplyViewToListWebPart

 Microsoft.SharePoint.Library.SPRequest.ApplyWebTemplate

 Microsoft.SharePoint.Library.SPRequest.BackupSite

 Microsoft.SharePoint.Library.SPRequest.BreakRoleDefsInheritance

 Microsoft.SharePoint.Library.SPRequest.BuildCabinetFile

 Microsoft.SharePoint.Library.SPRequest.BypassUseRemoteApis

 Microsoft.SharePoint.Library.SPRequest.CalculatePermissionsForCurrentThread

 Microsoft.SharePoint.Library.SPRequest.CallCalcEngine

 Microsoft.SharePoint.Library.SPRequest.CancelMeeting

 Microsoft.SharePoint.Library.SPRequest.CancelWorkflow

 Microsoft.SharePoint.Library.SPRequest.ChangeAccountPassword

 Microsoft.SharePoint.Library.SPRequest.CheckInFile

 Microsoft.SharePoint.Library.SPRequest.CheckOutFile

 Microsoft.SharePoint.Library.SPRequest.CheckZoneProps

 Microsoft.SharePoint.Library.SPRequest.ClearAllVars

 Microsoft.SharePoint.Library.SPRequest.ClearListCache

 Microsoft.SharePoint.Library.SPRequest.ClearTimerStoreServer

 Microsoft.SharePoint.Library.SPRequest.CloseStream

 Microsoft.SharePoint.Library.SPRequest.CompleteInProgressWorkItems

 Microsoft.SharePoint.Library.SPRequest.ConfirmUsage

 Microsoft.SharePoint.Library.SPRequest.CreateAuditEntry

 Microsoft.SharePoint.Library.SPRequest.CreateAuditEntryForUrl

 Microsoft.SharePoint.Library.SPRequest.CreateCustomList

 Microsoft.SharePoint.Library.SPRequest.CreateFolderOnImport

 Microsoft.SharePoint.Library.SPRequest.CreateList

 Microsoft.SharePoint.Library.SPRequest.CreateListFromFormPost

 Microsoft.SharePoint.Library.SPRequest.CreateListOnImport

 Microsoft.SharePoint.Library.SPRequest.CreateListViewPart

 Microsoft.SharePoint.Library.SPRequest.CreateOrUpdateFileAndItem

 Microsoft.SharePoint.Library.SPRequest.CreateSite

 Microsoft.SharePoint.Library.SPRequest.CreateView

 Microsoft.SharePoint.Library.SPRequest.CreateViewOnImport

 Microsoft.SharePoint.Library.SPRequest.CreateWeb

 Microsoft.SharePoint.Library.SPRequest.CrossListQuery

 Microsoft.SharePoint.Library.SPRequest.CustomizeCssFile

 Microsoft.SharePoint.Library.SPRequest.DeleteAllFileVersions

 Microsoft.SharePoint.Library.SPRequest.DeleteAllListItemVersions

 Microsoft.SharePoint.Library.SPRequest.DeleteCommentsOfDocs

 Microsoft.SharePoint.Library.SPRequest.DeleteFileVersion

 Microsoft.SharePoint.Library.SPRequest.DeleteInProgressWorkItems

 Microsoft.SharePoint.Library.SPRequest.DeleteItem

 Microsoft.SharePoint.Library.SPRequest.DeleteList

 Microsoft.SharePoint.Library.SPRequest.DeleteListItemVersion

 Microsoft.SharePoint.Library.SPRequest.DeleteNavigationNode

 Microsoft.SharePoint.Library.SPRequest.DeleteSite

 Microsoft.SharePoint.Library.SPRequest.DeleteSubscription

 Microsoft.SharePoint.Library.SPRequest.DeleteView

 Microsoft.SharePoint.Library.SPRequest.DeleteWeb

 Microsoft.SharePoint.Library.SPRequest.DeleteWebPart

 Microsoft.SharePoint.Library.SPRequest.DeleteWebPartPagePersonalization

 Microsoft.SharePoint.Library.SPRequest.DeleteWebPartPagePersonalizationForAUser

 Microsoft.SharePoint.Library.SPRequest.DeleteWebPartPersonalization

 Microsoft.SharePoint.Library.SPRequest.DeleteWorkflowAssociation

 Microsoft.SharePoint.Library.SPRequest.DeleteWorkItem

 Microsoft.SharePoint.Library.SPRequest.DetectOrphans

 Microsoft.SharePoint.Library.SPRequest.DispatchTimerJob

 Microsoft.SharePoint.Library.SPRequest.EnableModule

 Microsoft.SharePoint.Library.SPRequest.EnableModuleFromXml

 Microsoft.SharePoint.Library.SPRequest.EnsureSystemAccount

 Microsoft.SharePoint.Library.SPRequest.EnsureUserExists

 Microsoft.SharePoint.Library.SPRequest.ExecSiteSearch

 Microsoft.SharePoint.Library.SPRequest.ExecuteBatchReorder

 Microsoft.SharePoint.Library.SPRequest.ExpandListSchemaForExport

 Microsoft.SharePoint.Library.SPRequest.ExportNavigationXml

 Microsoft.SharePoint.Library.SPRequest.ExtractFilesFromCabinet

 Microsoft.SharePoint.Library.SPRequest.FetchActiveFeaturesFromSessionCache

 Microsoft.SharePoint.Library.SPRequest.FIrmProtectorFor

 Microsoft.SharePoint.Library.SPRequest.ForceDeleteList

 Microsoft.SharePoint.Library.SPRequest.FormatDateAsString

 Microsoft.SharePoint.Library.SPRequest.GenerateChangeNumber

 Microsoft.SharePoint.Library.SPRequest.GetAcceptHeaderExtensionsAsStringList

 Microsoft.SharePoint.Library.SPRequest.GetAclForCurrentWeb

 Microsoft.SharePoint.Library.SPRequest.GetAclForScope

 Microsoft.SharePoint.Library.SPRequest.GetAdminRecycleBinItems

 Microsoft.SharePoint.Library.SPRequest.GetAdminRecycleBinItemsForUI

 Microsoft.SharePoint.Library.SPRequest.GetAdminRecycleBinStatistics

 Microsoft.SharePoint.Library.SPRequest.GetAllAclsForCurrentSite

 Microsoft.SharePoint.Library.SPRequest.GetAllAuthenticatedUsersString

 Microsoft.SharePoint.Library.SPRequest.GetAllRolesForCurrentUser

 Microsoft.SharePoint.Library.SPRequest.GetAllWebsOfSite

 Microsoft.SharePoint.Library.SPRequest.GetAttachmentsInfo

 Microsoft.SharePoint.Library.SPRequest.GetCachedNavigationData

 Microsoft.SharePoint.Library.SPRequest.GetColumnDistinctAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetContainingList

 Microsoft.SharePoint.Library.SPRequest.GetContentTypeSchema

 Microsoft.SharePoint.Library.SPRequest.GetContentTypeWorkflowAssociationsAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetContextEventReceivers

 Microsoft.SharePoint.Library.SPRequest.GetCurrentUserPermissionOnGroup

 Microsoft.SharePoint.Library.SPRequest.GetCustomizedDocumentsInWeb

 Microsoft.SharePoint.Library.SPRequest.GetDbNow

 Microsoft.SharePoint.Library.SPRequest.GetDeadWebInfo

 Microsoft.SharePoint.Library.SPRequest.GetDocEventReceivers

 Microsoft.SharePoint.Library.SPRequest.GetDocsHavingComments

 Microsoft.SharePoint.Library.SPRequest.GetEffectiveRightsForCurrentUser

 Microsoft.SharePoint.Library.SPRequest.GetEventReceivers

 Microsoft.SharePoint.Library.SPRequest.GetExecuteUrl

 Microsoft.SharePoint.Library.SPRequest.GetExpandedFolderNSPath

 Microsoft.SharePoint.Library.SPRequest.GetExternalSecurityProviderConfiguration

 Microsoft.SharePoint.Library.SPRequest.GetExternalSecurityProviderId

 Microsoft.SharePoint.Library.SPRequest.GetFieldsSchemaXml

 Microsoft.SharePoint.Library.SPRequest.GetFieldTypeInfo

 Microsoft.SharePoint.Library.SPRequest.GetFile

 Microsoft.SharePoint.Library.SPRequest.GetFileAndFolderProperties

 Microsoft.SharePoint.Library.SPRequest.GetFileAndFpMetaInfo

 Microsoft.SharePoint.Library.SPRequest.GetFileAndMetaInfo

 Microsoft.SharePoint.Library.SPRequest.GetFileAsByteArray

 Microsoft.SharePoint.Library.SPRequest.GetFileAsStream

 Microsoft.SharePoint.Library.SPRequest.GetFileUrlFromViewTitle

 Microsoft.SharePoint.Library.SPRequest.GetFileVersionAsByteArray

 Microsoft.SharePoint.Library.SPRequest.GetFileVersions

 Microsoft.SharePoint.Library.SPRequest.GetFirstUniqueAncestorWebUrl

 Microsoft.SharePoint.Library.SPRequest.GetFolderContentTypeId

 Microsoft.SharePoint.Library.SPRequest.GetFolderContentTypeOrder

 Microsoft.SharePoint.Library.SPRequest.GetGhostedFile

 Microsoft.SharePoint.Library.SPRequest.GetGlobalContentTypeXml

 Microsoft.SharePoint.Library.SPRequest.GetGroupsDataAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetHash

 Microsoft.SharePoint.Library.SPRequest.GetHtmlTrCacheItemIfValid

 Microsoft.SharePoint.Library.SPRequest.GetHtmlTrUrlFromExt

 Microsoft.SharePoint.Library.SPRequest.GetIcon

 Microsoft.SharePoint.Library.SPRequest.GetIgnoreCanary

 Microsoft.SharePoint.Library.SPRequest.GetInfoFromCabinet

 Microsoft.SharePoint.Library.SPRequest.GetLanguageAttributes

 Microsoft.SharePoint.Library.SPRequest.GetLinkInfo

 Microsoft.SharePoint.Library.SPRequest.GetListAndChildrenNSInfo

 Microsoft.SharePoint.Library.SPRequest.GetListContentTypes

 Microsoft.SharePoint.Library.SPRequest.GetListCurrentFolderInfo

 Microsoft.SharePoint.Library.SPRequest.GetListItemDataAndRenderedViewWithCallback

 Microsoft.SharePoint.Library.SPRequest.GetListItemDataWithCallback

 Microsoft.SharePoint.Library.SPRequest.GetListItemPerm

 Microsoft.SharePoint.Library.SPRequest.GetListItemWorkflowAsSafeArrayAndLock

 Microsoft.SharePoint.Library.SPRequest.GetListItemWorkflowsAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetListScopeDataAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetListsWithCallback

 Microsoft.SharePoint.Library.SPRequest.GetListTemplates

 Microsoft.SharePoint.Library.SPRequest.GetListWorkflowAssociationsAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetMetadataForUrl

 Microsoft.SharePoint.Library.SPRequest.GetMinFieldTypeInfo

 Microsoft.SharePoint.Library.SPRequest.GetModules

 Microsoft.SharePoint.Library.SPRequest.GetMtgInstanceID

 Microsoft.SharePoint.Library.SPRequest.GetMtgResponseCookie

 Microsoft.SharePoint.Library.SPRequest.GetNavigationNode

 Microsoft.SharePoint.Library.SPRequest.GetNavigationNodeChild

 Microsoft.SharePoint.Library.SPRequest.GetNavigationNodeProperties

 Microsoft.SharePoint.Library.SPRequest.GetNTFullNameandEmailfromLogin

 Microsoft.SharePoint.Library.SPRequest.GetNTFullNamefromLogin

 Microsoft.SharePoint.Library.SPRequest.GetNTFullNamefromLoginEx

 Microsoft.SharePoint.Library.SPRequest.GetPageListId

 Microsoft.SharePoint.Library.SPRequest.GetParentWebUrl

 Microsoft.SharePoint.Library.SPRequest.GetPermissionsDataAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetPortalServerSettings

 Microsoft.SharePoint.Library.SPRequest.GetPortalSubscriptionUrl

 Microsoft.SharePoint.Library.SPRequest.GetPropertiesXmlForUncustomizedViews

 Microsoft.SharePoint.Library.SPRequest.GetRarelyUsedWebProps

 Microsoft.SharePoint.Library.SPRequest.GetRecycleBinItems

 Microsoft.SharePoint.Library.SPRequest.GetRecycleBinItemsForUI

 Microsoft.SharePoint.Library.SPRequest.GetRoleDefsDataAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetRunnableWorkItemsAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetSchemaXML

 Microsoft.SharePoint.Library.SPRequest.GetSecurityInfo

 Microsoft.SharePoint.Library.SPRequest.GetServerFileRedirect

 Microsoft.SharePoint.Library.SPRequest.GetSingleViewSchemaXml

 Microsoft.SharePoint.Library.SPRequest.GetSiteFlags

 Microsoft.SharePoint.Library.SPRequest.GetSiteItemSizes

 Microsoft.SharePoint.Library.SPRequest.GetSiteQuota

 Microsoft.SharePoint.Library.SPRequest.GetSiteUsageSummary

 Microsoft.SharePoint.Library.SPRequest.GetSizeOfWebPartsOnPage

 Microsoft.SharePoint.Library.SPRequest.GetStackTraceOnCreate

 Microsoft.SharePoint.Library.SPRequest.GetSTSVersion

 Microsoft.SharePoint.Library.SPRequest.GetSubscriptions

 Microsoft.SharePoint.Library.SPRequest.GetSubwebs

 Microsoft.SharePoint.Library.SPRequest.GetSubwebsFiltered

 Microsoft.SharePoint.Library.SPRequest.GetTimerRunningJobs

 Microsoft.SharePoint.Library.SPRequest.GetTimeZoneInfo

 Microsoft.SharePoint.Library.SPRequest.GetTimeZoneMoveParameters

 Microsoft.SharePoint.Library.SPRequest.GetTokenOfCurrentUser

 Microsoft.SharePoint.Library.SPRequest.GetUncustomizedDefaultView

 Microsoft.SharePoint.Library.SPRequest.GetUncustomizedViewByBaseViewId

 Microsoft.SharePoint.Library.SPRequest.GetUserAccountDirectoryPath

 Microsoft.SharePoint.Library.SPRequest.GetUserRegionalSettings

 Microsoft.SharePoint.Library.SPRequest.GetUsersDataAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetUserStorageInfo

 Microsoft.SharePoint.Library.SPRequest.GetUserToken

 Microsoft.SharePoint.Library.SPRequest.GetVersionIndependentProp

 Microsoft.SharePoint.Library.SPRequest.GetVersionIndependentProps

 Microsoft.SharePoint.Library.SPRequest.GetViewsSchemaXml

 Microsoft.SharePoint.Library.SPRequest.GetViewStylesXML

 Microsoft.SharePoint.Library.SPRequest.GetWebAncestry

 Microsoft.SharePoint.Library.SPRequest.GetWebAndChildrenNSInfo

 Microsoft.SharePoint.Library.SPRequest.GetWebListPermMask

 Microsoft.SharePoint.Library.SPRequest.GetWebMetainfo

 Microsoft.SharePoint.Library.SPRequest.GetWebPartPagePersonalizations

 Microsoft.SharePoint.Library.SPRequest.GetWebSubscriptionsUniqueUsers

 Microsoft.SharePoint.Library.SPRequest.GetWebsWithTimestamps

 Microsoft.SharePoint.Library.SPRequest.GetWebTemplates

 Microsoft.SharePoint.Library.SPRequest.GetWebThemeComposite

 Microsoft.SharePoint.Library.SPRequest.GetWebUrl

 Microsoft.SharePoint.Library.SPRequest.GetWebUsageData

 Microsoft.SharePoint.Library.SPRequest.GetWorkflowAssociationAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetWorkflowDataForItemAsSafeArrays

 Microsoft.SharePoint.Library.SPRequest.GetWorkItemAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GetWorkItemsAsSafeArray

 Microsoft.SharePoint.Library.SPRequest.GregorianISOToIntlISODate

 Microsoft.SharePoint.Library.SPRequest.HandleCookieOrAcceptTypes

 Microsoft.SharePoint.Library.SPRequest.ImportNavigationXml

 Microsoft.SharePoint.Library.SPRequest.InitHeap

 Microsoft.SharePoint.Library.SPRequest.InsertAlertEvent

 Microsoft.SharePoint.Library.SPRequest.InvalidateWebIdCache

 Microsoft.SharePoint.Library.SPRequest.InvokeTimerJob

 Microsoft.SharePoint.Library.SPRequest.IrmClientPresent

 Microsoft.SharePoint.Library.SPRequest.IrmClientReady

 Microsoft.SharePoint.Library.SPRequest.IsAttachment

 Microsoft.SharePoint.Library.SPRequest.IsCurrentUserMachineAdmin

 Microsoft.SharePoint.Library.SPRequest.IsCurrentUserMemberOfGroup

 Microsoft.SharePoint.Library.SPRequest.IsCurrentUserSiteAdmin

 Microsoft.SharePoint.Library.SPRequest.IsUrlSafeForRedirect

 Microsoft.SharePoint.Library.SPRequest.IsValidLoginName

 Microsoft.SharePoint.Library.SPRequest.IsVotingAllowed

 Microsoft.SharePoint.Library.SPRequest.ListRegionalOptions

 Microsoft.SharePoint.Library.SPRequest.LocalizeText

 Microsoft.SharePoint.Library.SPRequest.LocalizeXml

 Microsoft.SharePoint.Library.SPRequest.LogBinary

 Microsoft.SharePoint.Library.SPRequest.LogHelper

 Microsoft.SharePoint.Library.SPRequest.ManageAdminRecycleBin

 Microsoft.SharePoint.Library.SPRequest.ManageRecycleBin

 Microsoft.SharePoint.Library.SPRequest.MapUrlToListAndView

 Microsoft.SharePoint.Library.SPRequest.MigrateUserAccount

 Microsoft.SharePoint.Library.SPRequest.MiniSproc

 Microsoft.SharePoint.Library.SPRequest.ModifySubscription

 Microsoft.SharePoint.Library.SPRequest.MoveNavigationNode

 Microsoft.SharePoint.Library.SPRequest.MoveUrl

 Microsoft.SharePoint.Library.SPRequest.NavStructContainsPage

 Microsoft.SharePoint.Library.SPRequest.OpenSite

 Microsoft.SharePoint.Library.SPRequest.OpenWeb

 Microsoft.SharePoint.Library.SPRequest.OpenWebInternal

 Microsoft.SharePoint.Library.SPRequest.ParseMetaInfo

 Microsoft.SharePoint.Library.SPRequest.PatchUrl

 Microsoft.SharePoint.Library.SPRequest.PreInitServer

 Microsoft.SharePoint.Library.SPRequest.ProcessBatchData

 Microsoft.SharePoint.Library.SPRequest.PublishFile

 Microsoft.SharePoint.Library.SPRequest.PutFile

 Microsoft.SharePoint.Library.SPRequest.QueryGroups

 Microsoft.SharePoint.Library.SPRequest.QueryGroupsByIds

 Microsoft.SharePoint.Library.SPRequest.QueryUserInfo

 Microsoft.SharePoint.Library.SPRequest.ReadAuditFlags

 Microsoft.SharePoint.Library.SPRequest.ReadPagesRightsCheck

 Microsoft.SharePoint.Library.SPRequest.RecalculateSiteDiskUsed

 Microsoft.SharePoint.Library.SPRequest.ReCalculateWebFGP

 Microsoft.SharePoint.Library.SPRequest.RegisterContextEventReceiver

 Microsoft.SharePoint.Library.SPRequest.RegisterDocEventReceiver

 Microsoft.SharePoint.Library.SPRequest.RegisterEventReceiver

 Microsoft.SharePoint.Library.SPRequest.ReleaseResources

 Microsoft.SharePoint.Library.SPRequest.RelinkMeeting

 Microsoft.SharePoint.Library.SPRequest.RemoveExternalSecurityProvider

 Microsoft.SharePoint.Library.SPRequest.RemoveField

 Microsoft.SharePoint.Library.SPRequest.RemoveGroup

 Microsoft.SharePoint.Library.SPRequest.RemoveRoleDef

 Microsoft.SharePoint.Library.SPRequest.RemoveWorkflowFromListItem

 Microsoft.SharePoint.Library.SPRequest.RenameWeb

 Microsoft.SharePoint.Library.SPRequest.RenderColumn

 Microsoft.SharePoint.Library.SPRequest.RenderErrorPage

 Microsoft.SharePoint.Library.SPRequest.RenderFormAsHtml

 Microsoft.SharePoint.Library.SPRequest.RenderFormDigest

 Microsoft.SharePoint.Library.SPRequest.RenderListProperty

 Microsoft.SharePoint.Library.SPRequest.RenderListRelatedTasks

 Microsoft.SharePoint.Library.SPRequest.RenderNavigationBar

 Microsoft.SharePoint.Library.SPRequest.RenderSearchForm

 Microsoft.SharePoint.Library.SPRequest.RenderViewAsHtml

 Microsoft.SharePoint.Library.SPRequest.RepairOrphans

 Microsoft.SharePoint.Library.SPRequest.ReserveItemIdsForWorkflow

 Microsoft.SharePoint.Library.SPRequest.ResetSecurityScope

 Microsoft.SharePoint.Library.SPRequest.RestoreAdminRecycleBinItem

 Microsoft.SharePoint.Library.SPRequest.RestoreFileVersion

 Microsoft.SharePoint.Library.SPRequest.RestoreRecycleBinItem

 Microsoft.SharePoint.Library.SPRequest.RestoreSite

 Microsoft.SharePoint.Library.SPRequest.RevertContentStreams

 Microsoft.SharePoint.Library.SPRequest.RevertInProgressWorkItem

 Microsoft.SharePoint.Library.SPRequest.RevertInProgressWorkItems

 Microsoft.SharePoint.Library.SPRequest.SaveListAsTemplate

 Microsoft.SharePoint.Library.SPRequest.SaveWebAsTemplate

 Microsoft.SharePoint.Library.SPRequest.SealField

 Microsoft.SharePoint.Library.SPRequest.SeedEtag

 Microsoft.SharePoint.Library.SPRequest.SelectSitesAndUserInfoForMigration

 Microsoft.SharePoint.Library.SPRequest.SendMail

 Microsoft.SharePoint.Library.SPRequest.SetAnonymousAccessMask

 Microsoft.SharePoint.Library.SPRequest.SetAttendeeResponse

 Microsoft.SharePoint.Library.SPRequest.SetAuditFlags

 Microsoft.SharePoint.Library.SPRequest.SetDisableAsyncEvents

 Microsoft.SharePoint.Library.SPRequest.SetExactWebUrlFlag

 Microsoft.SharePoint.Library.SPRequest.SetGhostedFile

 Microsoft.SharePoint.Library.SPRequest.SetHtmlTrCacheItem

 Microsoft.SharePoint.Library.SPRequest.SetHttpParameters

 Microsoft.SharePoint.Library.SPRequest.SetIgnoreCanary

 Microsoft.SharePoint.Library.SPRequest.SetIgnoreCheckoutLock

 Microsoft.SharePoint.Library.SPRequest.SetIPAddr

 Microsoft.SharePoint.Library.SPRequest.SetListContentTypes

 Microsoft.SharePoint.Library.SPRequest.SetListProps

 Microsoft.SharePoint.Library.SPRequest.SetMondoProcHint

 Microsoft.SharePoint.Library.SPRequest.SetMtgInstanceID

 Microsoft.SharePoint.Library.SPRequest.SetPortalServerSettings

 Microsoft.SharePoint.Library.SPRequest.SetRequestAccessInfo

 Microsoft.SharePoint.Library.SPRequest.SetSiteFlags

 Microsoft.SharePoint.Library.SPRequest.SetSiteProps

 Microsoft.SharePoint.Library.SPRequest.SetSiteQuota

 Microsoft.SharePoint.Library.SPRequest.SetUserAccountDirectoryPath

 Microsoft.SharePoint.Library.SPRequest.SetVar

 Microsoft.SharePoint.Library.SPRequest.SetVersionIndependentProps

 Microsoft.SharePoint.Library.SPRequest.SetVersionIndependentPropsAdditive

 Microsoft.SharePoint.Library.SPRequest.SetWebAssociatedGroups

 Microsoft.SharePoint.Library.SPRequest.SetWebMetainfo

 Microsoft.SharePoint.Library.SPRequest.SetWebProps

 Microsoft.SharePoint.Library.SPRequest.SscCreateSite

 Microsoft.SharePoint.Library.SPRequest.TakePublishFileOffline

 Microsoft.SharePoint.Library.SPRequest.ThrowError

 Microsoft.SharePoint.Library.SPRequest.TZConvertDate

 Microsoft.SharePoint.Library.SPRequest.UncheckOutFile

 Microsoft.SharePoint.Library.SPRequest.UpdateField

 Microsoft.SharePoint.Library.SPRequest.UpdateFileOrFolderProperties

 Microsoft.SharePoint.Library.SPRequest.UpdateGroup

 Microsoft.SharePoint.Library.SPRequest.UpdateListItemWorkflow

 Microsoft.SharePoint.Library.SPRequest.UpdateListItemWorkflowLock

 Microsoft.SharePoint.Library.SPRequest.UpdateListSecurityTrim

 Microsoft.SharePoint.Library.SPRequest.UpdateMeeting

 Microsoft.SharePoint.Library.SPRequest.UpdateMeetingFromICal

 Microsoft.SharePoint.Library.SPRequest.UpdateMembers

 Microsoft.SharePoint.Library.SPRequest.UpdateNavigationNode

 Microsoft.SharePoint.Library.SPRequest.UpdateRoleAssignment

 Microsoft.SharePoint.Library.SPRequest.UpdateRoleDef

 Microsoft.SharePoint.Library.SPRequest.UpdateSiteHashKey

 Microsoft.SharePoint.Library.SPRequest.UpdateTimerRunningJobProgress

 Microsoft.SharePoint.Library.SPRequest.UpdateUser

 Microsoft.SharePoint.Library.SPRequest.UpdateView

 Microsoft.SharePoint.Library.SPRequest.UpdateWebPar

 Microsoft.SharePoint.Library.SPRequest.UpdateWebPartCache

 Microsoft.SharePoint.Library.SPRequest.UpdateWebPartIsIncluded

 Microsoft.SharePoint.Library.SPRequest.UpdateWebPartTypeId

 Microsoft.SharePoint.Library.SPRequest.UpdateWorkflowAssociation

 Microsoft.SharePoint.Library.SPRequest.UpdateWorkItem

 Microsoft.SharePoint.Library.SPRequest.UseDefaultAssociatedGroups

 Microsoft.SharePoint.Library.SPRequest.ValidateFormDigest

 Microsoft.SharePoint.Library.SPRequest.ValidateSubscriptionFilter

 Microsoft.SharePoint.Library.SPRequest.VssHelperIsEnabled

 Microsoft.SharePoint.Library.SPRequest.VssHelperOnIdentify

 Microsoft.SharePoint.Library.SPRequest.VssHelperOnPostRestore

 Microsoft.SharePoint.Library.SPRequest.VssHelperOnPreBackup

 Microsoft.SharePoint.Library.SPRequest.WebTemplateName

For SharePoint, Help Function for Site Name Retrieving

For System Center 2012 SP1 these resources are included:

 Microsoft.SharePoint.WebPartPages.SPWebPartManager.LoadWebParts

For SharePoint, KPI Helper

For System Center 2012 SP1 these resources are included:

 Microsoft.SharePoint.Portal.WebControls.KpiRenderer.getKpiData

For MVC

For System Center 2012 SP1 these resources are included:

 System.Web.Mvc.ReflectedActionDescriptor.Execute

 System.Web.Mvc.Async.ReflectedAsyncActionDescriptor.BeginExecute

For Windows Azure Storage Queue

For System Center 2012 SP1 these resources are included:

 Microsoft.WindowsAzure.StorageClient.CloudQueue.AddMessage

 Microsoft.WindowsAzure.StorageClient.CloudQueue.Clear

 Microsoft.WindowsAzure.StorageClient.CloudQueue.Create

 Microsoft.WindowsAzure.StorageClient.CloudQueue.CreateIfNotExist

 Microsoft.WindowsAzure.StorageClient.CloudQueue.Delete

 Microsoft.WindowsAzure.StorageClient.CloudQueue.UpdateMessage

 Microsoft.WindowsAzure.StorageClient.CloudQueue.DeleteMessage

 Microsoft.WindowsAzure.StorageClient.CloudQueue.Exists

 Microsoft.WindowsAzure.StorageClient.CloudQueue.GetMessagesInternal

 Microsoft.WindowsAzure.StorageClient.CloudQueue.PeekMessage

 Microsoft.WindowsAzure.StorageClient.CloudQueue.PeekMessages

 Microsoft.WindowsAzure.StorageClient.CloudQueue.FetchAttributes

 Microsoft.WindowsAzure.StorageClient.CloudQueue.SetMetadata

For Windows Azure Storage Table

For System Center 2012 SP1 these resources are included:

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.CreateTablesFromModel

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.CreateTable

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.CreateTableIfNotExist

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.DeleteTable

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.DeleteTableIfExist

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.DoesTableExist

 Microsoft.WindowsAzure.StorageClient.CloudTableClient.ListTablesSegmented

For Windows Azure Storage Blob Container

For System Center 2012 SP1 these resources are included:

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.Create

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.CreateIfNotExist

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.Delete

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.SetPermissions

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.GetPermissions

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.FetchAttributes

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.SetMetadata

 Microsoft.WindowsAzure.StorageClient.CloudBlobContainer.ListBlobsSegmented

 Microsoft.WindowsAzure.StorageClient.CloudBlobDirectory.ListBlobsSegmented

For Windows Azure Storage Blobs

For System Center 2012 SP1 these resources are included:

 Microsoft.WindowsAzure.StorageClient.CloudBlob.CopyFromBlob

 Microsoft.WindowsAzure.StorageClient.CloudBlob.CreateSnapshot

 Microsoft.WindowsAzure.StorageClient.CloudBlob.Delete

 Microsoft.WindowsAzure.StorageClient.CloudBlob.DeleteIfExists

 Microsoft.WindowsAzure.StorageClient.CloudBlob.UploadText

 Microsoft.WindowsAzure.StorageClient.CloudBlob.DownloadText

 Microsoft.WindowsAzure.StorageClient.CloudBlob.UploadFile

 Microsoft.WindowsAzure.StorageClient.CloudBlob.DownloadToFile

 Microsoft.WindowsAzure.StorageClient.CloudBlob.UploadFromStream

 Microsoft.WindowsAzure.StorageClient.CloudBlob.DownloadToStream

 Microsoft.WindowsAzure.StorageClient.CloudBlob.SetProperties

 Microsoft.WindowsAzure.StorageClient.CloudBlob.GetSharedAccessSignature

 Microsoft.WindowsAzure.StorageClient.CloudBlockBlob.PutBlock

 Microsoft.WindowsAzure.StorageClient.CloudBlockBlob.PutBlockList

 Microsoft.WindowsAzure.StorageClient.CloudBlockBlob.DownloadBlockList

 Microsoft.WindowsAzure.StorageClient.CloudPageBlob.WritePages

 Microsoft.WindowsAzure.StorageClient.CloudPageBlob.Create

 Microsoft.WindowsAzure.StorageClient.CloudPageBlob.ClearPages

 Microsoft.WindowsAzure.StorageClient.CloudPageBlob.GetPageRanges

For Windows Azure Storage Blob Stream

For System Center 2012 SP1 these resources are included:

 Microsoft.WindowsAzure.StorageClient.BlobReadStream.Read

 Microsoft.WindowsAzure.StorageClient.BlobWriteStream.Write

 Microsoft.WindowsAzure.StorageClient.BlobWriteStream.Commit

 Microsoft.WindowsAzure.StorageClient.CloudBlobClient.ListBlobsWithPrefixSegmented

 Microsoft.WindowsAzure.StorageClient.CloudBlobClient.ListContainersSegmented

 Microsoft.WindowsAzure.StorageClient.Tasks.Task`1.ExecuteAndWait

 System.Data.Services.Client.DataServiceContext.SaveChanges

 System.Data.Services.Client.DataServiceContext.Execute

 System.Data.Services.Client.DataServiceContext.ExecuteBatch

 Microsoft.WindowsAzure.StorageClient.TableServiceContext.SaveChangesWithRetries

For COM+ Services (Client Side)

 System.EnterpriseServices.RemoteServicedComponentProxy.Invoke

For COM+ Services (Server Side)

 For System Center 2012 SP1 only: System.EnterpriseServices.ServicedComponentProxy.LocalInvoke

For SQL server

 For System Center 2012 SP1 only: System.Data.SqlClient.SqlConnection.Open

 For System Center 2012 SP1 only: System.Data.SqlClient.SqlConnection.Close

 System.Data.SqlClient.SqlCommand.ExecuteReader

 System.Data.SqlClient.SqlCommand.ExecuteNonQuery

 System.Data.SqlClient.SqlCommand.ExecuteScalar

 System.Data.SqlClient.SqlDataAdapter.Fill

For OLEDB

 For System Center 2012 SP1 only: System.Data.OleDb.OleDbConnection.Open

 For System Center 2012 SP1 only: System.Data.OleDb.OleDbConnection.Close

 System.Data.OleDb.OleDbCommand.ExecuteReader

 System.Data.OleDb.OleDbCommand.System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.ExecuteReader

 System.Data.OleDb.OleDbCommand.ExecuteNonQuery

 System.Data.OleDb.OleDbCommand.ExecuteScalar

 System.Data.OleDb.OleDbDataAdapter.Fill

For ODBC

 For System Center 2012 SP1 only: System.Data.Odbc.OdbcConnection.Open

 For System Center 2012 SP1 only: System.Data.Odbc.OdbcConnection.Close

 System.Data.Odbc.OdbcCommand.ExecuteReader

 System.Data.Odbc.OdbcCommand.ExecuteNonQuery

 System.Data.Odbc.OdbcCommand.ExecuteScalar

 System.Data.Odbc.OdbcDataAdapter.Fill

For .NET Framework Data Provider for Oracle

 System.Data.OracleClient.OracleCommand.ExecuteNonQuery

 System.Data.OracleClient.OracleCommand.ExecuteReader

 System.Data.OracleClient.OracleCommand.ExecuteScalar

 System.Data.SqlClient.SqlConnection.Open

 System.Data.Odbc.OdbcConnection.Open

 System.Data.OleDb.OleDbConnection.Open

For Oracle

 For System Center 2012 SP1 only: System.Data.OracleClient.OracleConnection.Open

 For System Center 2012 SP1 only: System.Data.OracleClient.OracleConnection.Close

 Oracle.DataAccess.Client.OracleCommand.ExecuteReader

 Oracle.DataAccess.Client.OracleCommand.ExecuteNonQuery

 Oracle.DataAccess.Client.OracleCommand.ExecuteScalar

 System.Data.Common.DbAdapter.Fill

For System.Web.Mail

 System.Web.Mail.SmtpMail@CdoSysHelper.Send

For Instrumentation

 System.Web.UI.Page.ProcessRequest

 System.Web.UI.Page.ProcessRequest

 System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke

 System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke

 System.Web.Services.Protocols.LogicalMethodInfo.Invoke

For Instrumentation (Framework 1.0)

 System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse

For Instrumentation (Framework 1.1)

 System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse

 System.Web.HttpServerUtility.ExecuteInternal

 System.Web.UI.LosFormatter.Deserialize

For Instrumentation (Framework 2.0)

 System.Web.HttpServerUtility.Execute

For File System Operations

 System.IO.Directory.GetFileSystemEntries

 System.IO.Directory.GetFiles

 System.IO.File.Open

 System.IO.FileStream..ctor

 System.IO.FileStream.ReadCore

 System.IO.FileStream.WriteCore

 System.IO.FileStream.BeginReadCore

 System.IO.BeginWriteCore

 System.IO.Directory.InternalCreateDirectory

 System.IO.Directory.Delete

 System.IO.Directory.InternalGetFileDirectoryNames

 System.IO.DirectoryInfo.MoveTo

 System.IO.Directory.Move

 System.IO.File.Delete

 System.IO.FileInfo.Delete

 System.IO.File.Move

 System.IO.FileInfo.MoveTo

For Remoting Client-Side

 System.Runtime.Remoting.Proxies.RemotingProxy.InternalInvoke

 System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage

 System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke

For Remoting Server-Side

 System.Runtime.Remoting.Messaging.ServerObjectTerminatorSink.AsyncProcessMessage

 System.Runtime.Remoting.Messaging.ServerObjectTerminatorSink.SyncProcessMessage

For MS Message Queuing

 System.Messaging.MessageQueue.Send

 ystem.Messaging.MessageQueue.Receive

For Sockets

 System.NET.Sockets.Socket.Connect

 System.NET.Sockets.Socket.Receive

 System.NET.Sockets.Socket.Send

 System.NET.Sockets.Socket.Listen

For System.Web

 System.NET.FileWebRequest.GetResponse

 System.NET.FileWebRequest.GetRequestStream

 System.NET.HttpWebRequest.GetResponse

 System.NET.HttpWebRequest.GetRequestStream

 System.NET.FtpWebRequest.GetResponse

 System.NET.FtpWebRequest.GetRequestStream

 System.NET.WebClient.OpenRead

For Reflection

 System.Reflection.RuntimeConstructorInfo.Invoke

 System.RuntimeType.InvokeMember

For Registry

 Microsoft.Win32.RegistryKey.OpenSubKey

For Parsing

 System.Byte.Parse

 System.SByte.Parse

 System.Int16.Parse

 System.Int32.Parse

 System.Int64.Parse

 System.UInt16.Parse

 System.UInt32.Parse

 System.UInt64.Parse

 System.Single.Parse

 System.Double.Parse

 System.Boolean.Parse

 System.Char.Parse

 System.Decimal.Parse

 System.DateTime.Parse

 System.DateTime.ParseExact

 System.TimeSpan.Parse

 System.SqlBoolean.Parse

 System.SqlByte.Parse

 System.SqlDateTime.Parse

 System.SqlInt16.Parse

 System.SqlInt32.Parse

 System.SqlInt64.Parse

 System.SqlSingle.Parse

 System.SqlDouble.Parse

 System.SqlMoney.Parse

For IBM DB2

 IBM.Data.DB2.DB2Command.ExecuteReader

 IBM.Data.DB2.DB2Command.ExecuteReader

 IBM.Data.DB2.DB2Command.ExecuteNonQuery

 IBM.Data.DB2.DB2Command.ExecuteScalar

For Sybase

 Sybase.Data.AseClient.AseCommand.ExecuteScalarSybase.Data.AseClient.AseCommand.System.Data.IDbCommand.ExecuteReader

 Sybase.Data.AseClient.AseCommand.ExecuteNonQuery

Resource Calls Monitored by Default

Application Performance Monitoring includes well-known Microsoft .NET Framework resource calls that are monitored for slow performance.

For Custom Functions FW 1.0

 System.Data.Common.DbDataAdapter.FillFromReader

For Custom Functions FW 2.0

 System.Data.Common.DataAdapter.FillFromReader

For HTTP Handlers Calls .NET 1.1

 System.Web.Handlers.BatchHandler.System.Web.IHttpHandler.ProcessRequest

 System.Web.Handlers.TraceHandler.System.Web.IHttpHandler.ProcessRequest

 System.Web.HttpDebugHandler.ProcessRequest

 System.Web.HttpForbiddenHandler.ProcessRequest

 System.Web.HttpMethodNotAllowedHandler.ProcessRequest

 System.Web.HttpNotFoundHandler.ProcessRequest

 System.Web.HttpNotImplementedHandler.ProcessRequest

 System.Web.DefaultHttpHandler.ProcessRequest

 System.Web.SessionState.StateApplication.ProcessRequest

 System.Web.StaticFileHandler.ProcessRequest

 System.Web.UI.TrivialPage.ProcessRequest

For HTTP Handlers Calls .NET 2.0

 System.Web.Handlers.AssemblyResourceLoader.System.Web.IHttpHandler.ProcessRequest

For HTTP Modules Calls .NET 1.1

 System.Web.Caching.OutputCacheModule.OnEnter

 System.Web.Caching.OutputCacheModule.OnLeave

 System.Web.Security.DefaultAuthenticationModule.OnEnter

 System.Web.Security.FileAuthorizationModule.OnEnter

 System.Web.Security.FormsAuthenticationModule.OnAuthenticate

 System.Web.Security.PassportAuthenticationModule.OnAuthenticate

 System.Web.Security.UrlAuthorizationModule.OnEnter

 System.Web.Security.WindowsAuthenticationModule.OnAuthenticate

 System.Web.SessionState.SessionStateModule.BeginAcquireState

For HTTP Modules Calls .NET 2.0

 System.Web.Profile.ProfileModule.OnEnter

 System.Web.Profile.ProfileModule.OnLeave

 System.Web.Security.AnonymousIdentificationModule.OnEnter

 System.Web.Security.RoleManagerModule.OnEnter

 System.Web.UrlMappingsModule.OnEnter

For COM+ Server-Side

 System.EnterpriseServices.ServicedComponentProxy.LocalInvoke

For COM+ Client-Side

 System.EnterpriseServices.RemoteServicedComponentProxy.Invoke

For Remoting Client-Side

 System.Runtime.Remoting.Proxies.RemotingProxy.InternalInvoke

For Abstract Data Access Function

 System.Data.Common.DbDataAdapter.Update

For SQL Server Resources

 System.Data.SqlClient.SqlCommand.ExecuteReader

 System.Data.SqlClient.SqlCommand.ExecuteReader

 System.Data.SqlClient.SqlCommand.ExecuteXmlReader

 System.Data.SqlClient.SqlCommand.ExecuteNonQuery

 System.Data.SqlClient.SqlCommand.ExecuteScalar

 System.Data.SqlClient.SqlConnection.Open

 System.Data.SqlClient.SqlConnection.Close

For SQL Server Resources FW 2.0 Specific (Async Methods)

 System.Data.SqlClient.SqlCommand.BeginExecuteReader

 System.Data.SqlClient.SqlCommand.BeginExecuteNonQuery

For OLEDB Resources

 System.Data.OleDb.OleDbCommand.ExecuteReader

 System.Data.OleDb.OleDbCommand.ExecuteNonQuery

 System.Data.OleDb.OleDbCommand.ExecuteScalar

 System.Data.OleDb.OleDbConnection.Open

 System.Data.OleDb.OleDbConnection.Close

For ODBC Resources

 System.Data.Odbc.OdbcCommand.ExecuteReader

 System.Data.Odbc.OdbcCommand.ExecuteNonQuery

 System.Data.Odbc.OdbcCommand.ExecuteScalar

 System.Data.Odbc.OdbcConnection.Open

 System.Data.Odbc.OdbcConnection.Close

For Oracle Resources

 System.Data.OracleClient.OracleCommand.ExecuteReader

 System.Data.OracleClient.OracleCommand.ExecuteNonQuery

 System.Data.OracleClient.OracleCommand.ExecuteScalar

 System.Data.OracleClient.OracleConnection.Open

 System.Data.OracleClient.OracleConnection.Close

 Oracle.DataAccess.Client.OracleCommand.ExecuteReader

 Oracle.DataAccess.Client.OracleCommand.ExecuteNonQuery

 Oracle.DataAccess.Client.OracleCommand.ExecuteScalar

 Oracle.DataAccess.Client.OracleConnection.Open

 Oracle.DataAccess.Client.OracleConnection.Close

For IBM DB2 iSeries

 IBM.Data.DB2.iSeries.iDB2Command.ExecuteReader

 IBM.Data.DB2.iSeries.iDB2Command.ExecuteReader

 IBM.Data.DB2.iSeries.iDB2Command.ExecuteNonQuery

 IBM.Data.DB2.iSeries.iDB2Command.ExecuteScalar

 IBM.Data.DB2.iSeries.iDB2Connection.Open

 IBM.Data.DB2.iSeries.iDB2Connection.Close

For Web Services Resources

 System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke

 System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke

For System.Messaging

 System.Messaging.MessageQueue.Send

 System.Messaging.MessageQueue.Receive

For System.IO

 System.IO.File.Open

 System.IO.Directory.GetFileSystemEntries

 System.IO.Directory.GetFiles

For System.Web.Mail

 System.Web.Mail.SmtpMail.Send

For IBM DB2

 IBM.Data.DB2.DB2Command.ExecuteReader

 IBM.Data.DB2.DB2Command.ExecuteReader

 IBM.Data.DB2.DB2Command.ExecuteNonQuery

 IBM.Data.DB2.DB2Command.ExecuteScalar

 IBM.Data.DB2.DB2Connection.Open

 IBM.Data.DB2.DB2Connection.Close

For WCF Client-Side

 System.ServiceModel.Channels.ServiceChannelProxy.InvokeService

 System.ServiceModel.Channels.ServiceChannelProxy.InvokeBeginService

 System.ServiceModel.Channels.ServiceChannelProxy.InvokeEndService

For Mail

 System.Web.Mail.SmtpMail@CdoSysHelper.Send

For RK Sybase Resources

 Sybase.Data.AseClient.AseCommand.ExecuteScalar

 Sybase.Data.AseClient.AseCommand.ExecuteNonQuery

 Sybase.Data.AseClient.AseCommand.System.Data.IDbCommand.ExecuteReader

For LINQ

 System.Data.Linq.SqlClient.SqlProvider.System.Data.Linq.Provider.IProvider.Execute

 System.Data.Linq.SqlClient.SqlProvider.BuildQuery

 System.Data.Linq.SqlClient.SqlProvider.BuildQuery

For WWF

 System.Workflow.ComponentModel.ActivityExecutorDelegateInfo`1@ActivityExecutorDelegateOperation.Run

 System.Workflow.ComponentModel.ActivityExecutor`1.Execute

 System.Workflow.ComponentModel.Activity.SetStatus

 System.Workflow.ComponentModel.ActivityExecutionContext.ExecuteActivity

 System.Workflow.ComponentModel.Activity.Invoke

For ADOMD

 Microsoft.AnalysisServices.AdomdClient.AdomdCommand.ExecuteCellSet

 Microsoft.AnalysisServices.AdomdClient.AdomdCommand.ExecuteReader

 Microsoft.AnalysisServices.AdomdClient.AdomdCommand.ExecuteNonQuery

 Microsoft.AnalysisServices.AdomdClient.AdomdCommand.Execute

For AJAX.NET

 System.Web.Script.Services.RestHandler.InvokeMethod

For WebClient

 System.Net.WebClient.UploadBits

 System.Net.WebClient.DownloadBits

 System.Web.HttpPostedFile.SaveAs

For WebClient FW 1.1

 System.Net.WebClient.UploadValues System.Net.WebClient.UploadBits

 System.Net.WebClient.UploadFile

 System.Net.WebClient.DownloadFile

OLE DB Data Source Template

OLE DB (Object Linking and Embedding Database) is a Microsoft technology for accessing a variety of data sources by using a common method to connect to different databases such as Microsoft SQL Server.

The OLE DB Data Source template lets you monitor the availability and performance of any database that can be accessed with OLE DB. One or more watcher nodes connect to the database to verify its availability and test its performance. The watcher nodes can test the connection to the database or measure the time taken to perform a particular query.

The database can reside on any computer whether it has an agent for Operations Manager installed or not, but it must be accessible from the watcher nodes. Each watcher node must have an Operations Manager agent installed.

Scenarios

Use the OLE DB Data Source template in scenarios where applications rely on a database. You can either define a single watcher node to ensure that the database is accessible and responds to requests, or you can define each application server as a watcher node. The monitors that the template creates attempt to connect to the database from each location at the defined interval, and verify that each watcher node can connect successfully. In addition to validating the health of the database itself, any network connections and other required features between the watcher node and the database are also validated. You can use any number of watcher nodes, but it is typically most useful to select a sample that represents different environment or network segments.

Monitoring Performed by OLE DB Data Source Template

Depending on your selections in the OLE DB Data Source Template wizard, the monitoring performed by the created monitors and rules can include any of the following settings.

|  |  |  |
| --- | --- | --- |
| Type | Description | Enabled? |
| Monitors | Success of the database connection or query | Enabled by default. |
| Time to connect to database | Enabled if specified in wizard. |
| Time to complete query | Enabled if specified in the wizard and the query is provided. |
| Time to fetch results of query | Enabled if specified in the wizard and the query is provided. |
| Collection Rules | Collection of time to connect to the database | Enabled by default. |
| Collection of time to complete the query | Always enabled if the query is provided. |
| Collection of time to fetch results of query | Always enabled if the query is provided. |

Viewing Monitoring Data

All data collected by the OLE DB Data Source template is available in the OLE DB Data Source State view located in the Synthetic Transaction folder. In this view, an object represents each of the watcher nodes. The state of each object represents the worst state of the set of database monitors that are running on that node. If one or more of the nodes is shown with an error while at least one other node is healthy, it could indicate a problem with that particular node accessing the database, a network issue. If all of the nodes are unhealthy, it could indicate a problem with the database itself.

You can view the state of the individual database monitors by opening the Operations Manager Health Explorer for each object. You can view performance data by opening the Performance view for each of these objects.

Wizard Options

When you run the OLE DB Data Source template, you have to provide values for options in the following tables. Each table represents a single page in the wizard.

General Properties

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the monitoring wizard. This name is displayed in the Operations console in the OLE DB Data Source State view. |
| Description | Optional description of the monitor. |
| Management Pack | Management pack to store the classes, monitors, and rules that the template created.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Connection String

The following options are available on the Connection String page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Connection string | The connection string to connect to the database. A connection string contains the properties required to locate and connect to the database. It contains such information as the server that is hosting the database, the database name, and the type of authentication to perform. You can type a connection string or build a connection string in a dialog box by clicking Build.  For more detailed information about connection strings, see [Connection String Syntax](http://go.microsoft.com/fwlink/?LinkId=70748). |
| Query to execute | Optional query after the connection to the database has been made. If no query is provided, the monitor only attempts to connect to the database. |
| Query time-out | If a query is provided, this option specifies the maximum number of seconds that the query can take before it times out. You must set the Query Time-out value. |

Query Performance

The following options are available on the Query Performance page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Connection time in milliseconds | If selected, the values Error Threshold and Warning Threshold provide the number of milliseconds for the connection when the monitor enters that state and an alert is generated. If not selected, the connection time is not monitored. |
| Query time in milliseconds | If selected, the values Error Threshold and Warning Threshold define the number of milliseconds the query can run before the monitor enters that state and generates an alert. If not selected, the time to run the query is not monitored. If a query is not provided, this option is not available. |
| Fetch time in milliseconds | If selected, the values Error Threshold and Warning Threshold define the number of milliseconds to retrieve the results of the query before the monitor enters that state and generates an alert. If not selected, the time to retrieve the results of the query is not monitored. If a query is not provided, this option is not available. |

Watcher Nodes

The following options are available on the Watcher Nodes page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Select one or more agent-managed computers | Specify one or more agent-managed computers to run the monitor. For more information, see [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1). |
| Run this query every | The frequency to attempt the connection to the database and run the query, if specified. |

Security Considerations

The OLE DB Data Source template creates two Run As profiles. The name for each of these profiles starts with the name that you provided in the template and is followed by “Simple Authentication Profile” and “Synthetic Transaction Profile”. If no Run As account is added to either of these profiles, the Default Action Account for each watcher node is used to connect the database and run the query. If the Default Action Account does not have access to the database that is being monitored, the connection fails. You can specify either integrated security or simple authentication by creating a Run As account and adding it to the appropriate Run As profile that the OLE DB Data Source template created.

When you run the OLE DB Data Source template, it creates two Run As profiles. The name for each starts with the Name that you provided when you ran the template. The OLE DB Synthetic Transaction Profile is used when you want to use integrated security for the database connection. The OLE DB Simple Authentication Profile is used when you want to use simple authentication for the database connection.

Integrated Security

Integrated security lets you connect to the database by using credentials stored in Active Directory Domain Services. To connect the watcher nodes to the database by using integrated security, create a Run As account with Windows as the Account type and the credentials for the appropriate user account. Then add this Run As profile to the OLE DB Synthetic Transaction Profile.

Simple Authentication

Simple authentication lets you connect to the database by using a simple name and password. For a SQL Server database, this simple authentication could be used for SQL Server authentication. To have the watcher nodes connect to the database by using simple authentication, create a Run As account with Simple Authentication as the Account type and the credentials for the appropriate user account. Then add this Run As profile to the OLE DB Simple Authentication Profile. When you specify the connection string for the template, select the Use Simple Authentication RunAs Profile created for this OLE DB data source transaction check box. This adds variables to the connection string for the user name and password that you specified in the Run As account.

Creating and Modifying OLE DB Data Source Templates

To run the OLE DB Data Source wizard

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, right-click Management Pack Templates, and then select Add Monitoring Wizard.  4. On the Select Monitoring Type page, select OLE DB Data Source, and then click Next.  5. On the General Properties page, in the Name and Description boxes, type a name and an optional description.  6. Select a management pack in which to save the monitor, or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  7. Click Next.  8. In the Connection string box, type a connection string for the database, or click the Build button to be prompted for the required information.  9. If you want the monitor to run a query, select Query to execute, and then type a query.  10. If you want to set a time-out for the query, type the number of seconds in the Query time-out box.  11. Click Test to perform a test connection by using the connection string and query that you just provided.  Note  The test is performed on the workstation that you are using to run the template. If this workstation cannot access the database, this test fails. When the template is completed, the query is run from the watcher nodes that you specify.  12. Click Next when you have validated your connection string and query.  13. Select the measurements that you want to monitor and set an error and warning threshold for each. Click Next.  14. Select one or more [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1) to run the monitor.  15. Specify the frequency to run the monitor in the Run this query box. Click Next.  16. Review the summary of the monitor, and then click Create.  17. If a Run As account with credentials that have access to the database does not exist, create an appropriate Run As account in the Administration workspace. For more information, see How to Create a Run As Account.  Note  To create and modify a Run As account, you must have administrative credentials for the management group.  18. If the database uses integrated security, add the Run As account to the Synthetic Transaction Action Profile for the template. If the database uses simple authentication, add the Run As account to the Simple Authentication Profile for the template.  Note  To create and modify a Run As profile, you must have administrative credentials for the management group. |

To modify an existing OLE DB Data Source template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select OLE DB Data Source.  4. In the OLE DB Data Source pane, locate the monitor to change.  5. Right-click the monitor, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing OLE DB Data Source Monitors and Collected Data

To view all OLE DB Data Source monitors

|  |
| --- |
| 1. Open the Operations console.  2. Open the Monitoring workspace.  3. In the Monitoring navigation pane, select Synthetic Transaction, and then click OLE DB Data Source State. |

To view the state of each monitor

|  |
| --- |
| 1. In the OLE DB Data Source State pane, right-click an object. Select Open, and then click Health Explorer.  2. Expand the Availability and Performance nodes to view the individual monitors. |

To view the performance collected for a monitor

|  |
| --- |
| 1. In the OLE DB Data Source State pane, right-click an object. Select Open, and then click Performance.  2. In the Legend pane, select the counters that you want to view.  3. Use options in the Actions pane to modify the Performance view. |

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

[Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1)

Process Monitoring Template

The Process Monitoring template lets you monitor whether a particular process is running on a computer. By using this template, you can implement two different basic scenarios: You might require the process to be running for a particular application and want to be warned if it is not running, or you might have to be alerted if you discover that an unwanted process is running. In addition to monitoring whether the application is running, you can collect performance data for the processor and memory usage of the process.

Scenarios

Use the Process Monitoring template in different scenarios where you have to monitor a running process on an agent-managed Windows-based computer. Your application can monitor the following processes.

Critical Process

A process that must be running at all times. Use the Process Monitoring template to ensure that this process is running on the computers where it is installed, and use the Process Monitoring template to measure its performance.

Unwanted Process

A process that should not be running. This process might be a known rogue process that can cause damage, or it might be a process that is automatically started when an error in the application occurs. The Process Monitoring template can monitor for this process and send an alert if it is found to be running.

Long Running Process

A process that runs for short periods at a time. If the process is running for an excessive length of time, it might indicate a problem. The Process Monitoring template can monitor for the length of time that this process runs and send an alert if the running time exceeds a particular duration.

Monitoring Performed by Process Monitoring Template

Depending on your selections in the Process Monitoring wizard, the monitoring performed by the created monitors and rules can include any of the following settings.

|  |  |  |
| --- | --- | --- |
| Type | Description | When Enabled |
| Monitors | Count of wanted processes running | Enabled if you select Processes you want on the Process to Monitor page and Number of processes on the Running Processes page. |
| Time that a wanted process has been running | Enabled if you select Processes you want on the Process to Monitor page and Duration on the Running Processes page. |
| Unwanted process running | Enabled if Monitoring Scenario is for unwanted processes. |
| Processor utilization of process | Enabled if you select Processes you want on the Process to Monitor page, and you enable CPU alert on the Performance Data page. |
| Memory usage of process | Enabled if you select Processes you want on the Process to Monitor page, and you enable memory alert on the Performance Data page. |
| Collection Rules | Collection of processor utilization of process | Enabled if you select Processes you want on the Process to Monitor page, and you enable CPU alert on the Performance Data page. |
| Collection of memory usage of process. | Enabled if you select Processes you want on the Process to Monitor page, and you enable memory alert on the Performance Data page. |

Viewing Monitoring Data

All data collected by the Process Monitoring template is available in the Process State view located in the Windows Service and Process Monitoring folder. In this view, an object is listed for each agent in the group that you selected. Even if an agent does not monitor a process, it is listed, and the monitor reflects the state for the process that is not running.

You can view the state of the individual process monitors by opening the Operations Manager Health Explorer for the process object. You can view performance data by opening the Performance view for the process object.

The same process objects that are listed in the Process State view are included in the Health Explorer for the computer that hosts the process. The health state of the process monitors rolls up to the health of the computer.

Wizard Options

When you run the Process Monitoring template, you have to provide values for the options in the following tables. Each table represents a single page in the wizard.

General Properties

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the process. This name is displayed in the Operations console for the wizard. It does not have to be the same name as the process. |
| Description | Optional description of the process. |
| Management Pack | Management pack to store the class and monitors that the template creates. If you create any additional monitors or rules that are using the service as a target class, they have to be stored in the same management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Process to Monitor

The following options are available on the Process to Monitor page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Monitoring Scenario | The kind of monitoring that is to be performed. Select Monitor whether and how a process is running to monitor for a wanted process and set the monitor to a critical state when the process is not running. Select Monitor only whether a process is running to monitor for an unwanted process and set the monitor to a critical state when the process is running. |
| Process name | The full name of the process. This is the name of the process as it appears in Task Manager. It should not include the path to the actual executable file. You can either type the name or click the ellipse (…) button to locate the file name. |
| Targeted group | The process is monitored on all computers that are included in the specified group. |

Running Processes

The following options are available on the Running Processes page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Generate an alert of the number of processes is below the minimum value or above the maximum value for longer than the specified duration | If selected, the monitor is set to a critical state, and an alert is created if the number of instances of the specified process is less than the specified minimum or greater than the specified maximum for a longer period than the specified duration.  To ensure that at least one instance of the process is running, set both the minimum and maximum to 1. |
| Minimum number of processes | The minimum number of processes that should be running. |
| Maximum number of processes | The maximum number of processes that should be running. |
| Duration | Specifies how long the number of running processes must exceed the specified range before the monitor is set to a critical state. Do not set this value to less than 1 minute. |
| Generate an alert if the process runs longer than the specified duration | If selected, the monitor is set to a critical state, and an alert is created if one instance of the process runs for longer than the specified duration. |

Performance Data

The following options are available on the Performance Data page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Generate an alert if CPU usage exceeds the specified threshold | Specifies if CPU usage of the process should be monitored. A monitor will be created to set an error state on the object and generate an alert when the specified threshold is exceeded. A rule is created to collect CPU usage for analysis and reporting. |
| CPU Usage (percentage) | If CPU utilization is monitored, this option sets the threshold. If the percentage of total CPU usage exceeds the threshold, the object is set to an error state, and an alert is generated. |
| Generate an alert if memory usage exceeds the specified threshold | Specifies if memory usage of the process should be monitored. A monitor will be created to set an error state on the object, and generate an alert when the specified threshold is exceeded. A rule is created to collect CPU usage for analysis and reporting. |
| Memory Usage (MB) | If memory usage is monitored, this option sets the threshold. If the disk space in megabytes (MB) of total CPU usage exceeds the threshold, the object is set to an error state, and an alert is generated. |
| Number of samples | If CPU usage or memory is monitored, this option specifies the number of consecutive performance samples that must be exceeded before the object is set to an error state, and an alert is generated.  Specifying a number greater than 1 for this option limits the noise from monitoring by ensuring that an alert is not generated when the service only briefly exceeds the threshold. The larger the value that you set, the longer the period of time before you are alerted to a problem. A typical value is 2 or 3. |
| Sampling interval | If CPU usage or memory is monitored, specify the length of time between performance samples.  A smaller value for this option reduces the time for detecting a problem but increases overhead on the agent and the amount of data collected for reporting. A typical value is between 5 and 15 minutes. |

Additional Monitoring

In addition to performing the specified monitoring, the Process Monitoring template creates a targetd class that you can use for additional monitors and workflows. Any monitor or rule using this class as a target will run on any agent-managed computer in the group specified in the template. If it creates Windows events that indicate an error, for example, you could create a monitor or rule that detects the particular event and uses the process’ class as a target.

Creating and Modifying Process Monitor Templates

To run the Process Monitoring wizard

|  |
| --- |
| 1. Determine the target group for the monitor by using the following logic:   If you want to discover the process on all Windows-based computers in the management group, you do not have to create a group. You can use the existing group All Windows Computers.   If you only want the process to be discovered on a certain group of computers, either ensure that an appropriate group exists or create a new group by using the procedure in How to Create Groups in Operations Manager 2012.   If the process that you are monitoring is in a cluster, create a group with objects of the class Virtual Server representing the nodes of the cluster that contain the service.  2. Start the Add Monitoring wizard.  3. On the Select Monitoring Type page, select Process Monitoring, and then click Next.  4. On the General Properties page, in the Name and Description boxes, type a name and an optional description. The name is used to describe the process in the Operations console. It is not the actual name of the process.  5. Select a management pack in which to save the monitor, or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  6. Click Next.  7. On the Process to Monitor page, do the following:  a. Select whether you want to monitor a wanted or an unwanted process.  b. In the Process name box, type the complete name of the process to monitor. For example, notepad.exe. You can also click the ellipse (…) button and locate the executable file.  c. Click the ellipse (…) button to the right of the Targeted Group box, and then select the group from the first step of this procedure.  d. Click Next.  8. If you selected the option for a wanted process, on the Running Processes page, do the following:  a. If you want to monitor whether the process is running, do the following:  i. Select the option to Generate an alert of the number of processes is below the minimum value or above the maximum value for longer than the specified duration.  ii. In the Minimum number of processes box, enter the minimum number of processes that should be running. For a single instance of the process, this is typically 1.  iii. In the Maximum number of processes box, enter the maximum number of instances of the process that should be running.  iv. In the Duration box, enter the length of time that running processes must exceed the specified range before the monitor is set to a critical state. This value should not be set to less than 1 minute. Note that the process could stop and restart within this time window with no error detected.  b. If you want to monitor for the length that a process runs, do the following:  i. Select the option to Generate an alert if the process runs longer than the specified duration.  ii. In the Duration box, enter the maximum length of time that you want the process to run before the monitor is set to a critical state. This value should not be set to less than 1 minute.  9. If you selected the option for a wanted process, on the Performance Data page, select the performance counters and thresholds that you want to monitor. For more detailed information, see the [Wizard Options](#z44) section.  Note  This page is disabled if you selected the option for an unwanted process.  10. If you have selected performance counters, specify the monitoring interval.  11. Click Next.  12. Review the summary of the monitor, and then click Create. |

To modify an existing Process Monitoring template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then click Process Monitoring.  4. In the Process Monitoring pane, locate the monitor to change.  5. Right-click the monitor, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing Process Monitoring Monitors and Collected Data

To view all Process Monitoring monitors

|  |
| --- |
| 1. Open the Operations console.  2. Open the Monitoring workspace.  3. In the Monitoring navigation pane, select Windows Service and Process Monitoring, and then click Process State. |

To view the state of each monitor

|  |
| --- |
| 1. In the Process State pane, right-click an object. Select Open, and then click Health Explorer.  2. Expand the Availability and Performance nodes to view the individual monitors. |

To view the performance collected for a process

|  |
| --- |
| 1. In the Process State pane, right-click an object. Select Open, and then click Performance.  2. In the Legend pane, select the counters that you want to view.  3. Use options in the Actions pane to modify the Performance view. |

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

[Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1)

TCP Port Template

The TCP Port template lets you monitor the availability of an application that is accessible over TCP.

The application that is being tested can reside on any computer whether an agent for Operations Manager installed or not. Each watcher node must have an Operations Manager agent installed.

Scenarios

Use the TCP Port template in scenarios where applications rely on a service accessible over TCP. You can define each client as a watcher node. The monitors created by the template attempt to connect to the application from each client at the defined interval. The monitors verify that each client can connect successfully. In addition to validating the availability of the application itself, any network connections and other required features between the watcher node and the application are also validated.

Monitoring Performed by the TCP Port Template

The monitoring performed by the monitors and rules created by the TCP Port template can include any of the following settings.

|  |  |  |
| --- | --- | --- |
| Type | Description | Enabled? |
| Monitors | Target host reachable | Enabled |
| Connection accepted | Enabled |
| Connection timeout | Enabled |
| DNS resolution | Enabled |
| Collection Rules | Connection time | Enabled |

Viewing Monitoring Data

All data collected by the TCP Port template is available in the TCP Port Checks State view located in the Synthetic Transaction folder. In this view, an object represents each of the watcher nodes. The state of each object represents the worst state of the set of TCP Port monitors running on that node. If one or more of the nodes is shown with an error while at least one other node is healthy, it could indicate a problem with that particular node accessing the specified computer, for example, a network issue. If all of the nodes are unhealthy, it could indicate a problem with the application itself, either the computer being offline or the application not responding on the specified port.

You can view the state of the individual TCP Port monitors by opening the Operations Manager Health Explorer for each object. You can view performance data by opening the Performance view for each of these objects.

Wizard Options

When you run the TCP Port template, you have to provide values for options in the following tables. Each table represents a single page in the wizard.

General Options

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the template. This name is displayed in the Operations console. |
| Description | Optional description of the service. |
| Management Pack | Management pack to store the class and monitors that the template created. If you create any additional monitors or rules by using the service as a target class, you must store them in the same management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Target and Port

The following options are available on the Target and Port page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Computer or device name | The name of the computer or device to connect to. This can be a name or an IP address. It must be resolved by and accessible to each watcher node that you specify. |
| Port | The number of the port on which the application is listening. |

Watcher Nodes

The following options are available on the Watcher Nodes page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Select one or more agent managed computers | Specify one or more computers to run the monitors and rules. For more information, see [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1). |
| Run this query every | The frequency to attempt the connection to the specified computer and port. |

Creating and Modifying TCP Port Templates

To run the TCP Port data source wizard

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, right-click Management Pack Templates, and then select Add Monitoring Wizard.  4. On the Select Monitoring Type page, select TCP Port, and then click Next.  5. On the General Properties page, in the Name and Description boxes, type a name and an optional description.  6. Select a management pack in which to save the monitor, or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  7. Click Next.  8. In the Computer or device name box, type the name or IP address of the computer or device to connect to.  9. In the Port box, type the port number on which the application is listening.  10. Click Test to perform a test connection by using the connection string and query that you just provided.  Note  The test is performed on the workstation that you are using to run the template. If this workstation cannot access the computer or device, this test fails. When the template is completed, the test is run from the watcher nodes that you specify.  11. Click Next when you have validated your connection.  12. Select one or more [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1) to run the monitor.  13. Specify the frequency to run the monitor in the Run this query box. Click Next.  14. Review the summary of the monitor, and then click Create. |

To modify an existing TPC Port template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select TCP Port.  4. In the TCP Port pane, locate the template to change.  5. Right-click the monitor, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing TCP Port Monitors and Collected Data

To view all TCP Port monitors

|  |
| --- |
| 1. Open the Operations console.  2. Open the Monitoring workspace.  3. In the Monitoring navigation pane, select Synthetic Transaction, and then click TCP Port Checks State. |

To view the state of each monitor

|  |
| --- |
| 1. In the TCP Port Checks State pane, right-click an object. Select Open, and then click Health Explorer.  2. Expand the Availability and Performance nodes to view the individual monitors. |

To view the performance collected for a monitor

|  |
| --- |
| 1. In the TCP Port Checks State pane, right-click an object. Select Open, and then click Performance.  2. In the Legend pane, select the counters you want to view.  3. Use options in the Actions pane to modify the Performance view. |

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

[Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1)

UNIX or Linux Log File

The UNIX/Linux Log File Monitoring template lets you create an alert when a particular text is detected in a log file.

Scenarios

Use the UNIX/Linux Log File Monitoring template for any application that writes to a log file when a particular error occurs. You provide the path to the log file and the text that indicates an error, and an alert is created for you anytime that text is detected.

Monitoring Performed by the UNIX/Linux Log File Monitoring Template

The following table shows the monitoring activity that the UNIX/Linux Log FileMonitoring template performs.

|  |  |  |
| --- | --- | --- |
| Type | Description | When Enabled |
| Rule | Creates an alert when a specified text is detected. | Enabled |

Wizard Options

When you run the UNIX/Linux Log File Monitoring template, you have to provide values for the options in the following tables. Each table represents a single page in the wizard.

General Options

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the template. This name is displayed in the Operations console. |
| Description | Optional description of the template. |
| Management Pack | Management pack file to store the rule that the template creates.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Log File Details

The following options are available on the Log File Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Computer | If you want to monitor a single computer, enter the name of the agent-managed UNIX or Linux computer with the log file to monitor. Click the Select a Computer button to select one of the agent-managed UNIX or Linux computers that are installed in your management group. |
| Computer group name | If you are going to monitor a group of computers, enter the name of the group of agent-managed UNIX or Linux computers with the log file to monitor. Click the Computer Group button to select from the group in your management group. |
| Log file path | Complete path and name of the log file. |
| Expression | Regular expression of the text to detect. If you want to detect a simple string of characters, type the string of characters. |

Creating and Modifying UNIX/Linux Log File Templates

To create a UNIX/Linux Log File template

|  |
| --- |
| 1. If you want to monitor the log file on a group of computers, determine the target group for the monitor by using the following logic:   If you want to monitor the log file on all UNIX and Linux computers in the management group, you do not have to create a group. You can use the existing group UNIX/Linux Computer Group.   If you only want the log file to be monitored on a certain group of computers, either ensure that an appropriate group exists or create a new computer group by using the procedure in How to Create Groups in Operations Manager .  2. Start the Add Monitoring wizard.  3. On the Select Monitoring Type page, select UNIX/Linux Log File Monitoring, and then click Next.  4. On the General Properties page, in the Name and Description boxes, type a name and description for this new template.  5. Select a management pack in which to save the template or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  6. If you want to monitor the log file on a single computer, do the following:  a. Click the Select a Computer button next to the Computer name box.  b. Select the computer to monitor, and then click OK.  7. If you want to monitor the log file on a group of computers, do the following:  a. Click the Select a Group button next to the Computer group name box.  b. Select the computer to monitor, and then click OK.  8. In the Log file path box, type the path and name of the log file to monitor.  9. In the Expression box, type the text to watch for, such as error. You can type a regular expression for more complex logic.  10. Optionally, click Test to open a new dialog to test Regular expression matching against sample text that you input.  11. Select a Run As Profile to use. The account associated with the target computer in this profile will be used to read the log file.  12. Select an Alert Severity.  13. Click Next.  14. Verify the summary information for the template, and then click Create. |

To modify an existing UNIX/Linux Log File template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select UNIX/Linux Log File.  4. In the UNIX/Linux Log File Monitoring pane, locate the template to change.  5. Right-click the template, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing UNIX/Linux Log File Data

There is no monitor or collected data for the UNIX/Linux Log File Monitoring template. If a match is found in the specified log file, an alert is generated. You can view this alert in the Active Alerts view with the other alerts.

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

[Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1)

UNIX or Linux Process

The UNIX/Linux Process Monitoring template lets you monitor that a particular process installed on an UNIX or Linux computer runs.

Scenarios

The UNIX/Linux Process Monitoring template is useful for monitoring any application as monitoring processes is typically critical to the health of the application.

Monitoring Performed by the UNIX/Linux Process Monitoring Template

The following table shows the monitoring activity that the UNIX/Linux Process Monitoring template performs.

|  |  |  |
| --- | --- | --- |
| Type | Description | When Enabled |
| Monitors | Process Count is Outside of Range | Always enabled. |

Wizard Options

When you run the UNIX/Linux Process Monitoring template, you have to provide values for options in the following tables. Each table represents a single page in the wizard.

General Options

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the template. This is the name that is displayed in the Operations console. |
| Description | Optional description of the template. |
| Management Pack | Management pack to store the class and monitors that the template creates. If you create any additional monitors or rules that are using the process as a targeted process, you must store them in the same management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Process Monitoring Details

The following options are available on the Process Monitoring Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Process name | The name of the process. You can use the Select a Process button to connect to a monitored UNIX/Linux computer and list current running processes in order to select a process by name. If you wish to target the monitor to only a single computer, you must use the Select a Process button to select a computer and process. |
| Computer Group | The name of the group of UNIX or Linux computers for the process to monitor. Click the Select a group button to select a group that is installed in your management group. If you have used the Select a Process button to select a running process from a computer, the monitor will be targeted to that computer. After using the Select a Process button to select a process, you can use the Select a Group button to target a group with the monitor for the selected process. |
| Alert Severity | The severity for the alert: Error, Warning, or Information. |
| Regular Expression to filter process arguments | An optional Regular expression to use in filtering processes by arguments. If this option is used, processes that match the provided process name will be additionally filtered by their arguments. Only processes with arguments that match the Regular expression will be evaluated by the monitor. This is useful to identify a process for a specific application when other applications on the system may use a process with the same name. The Regular expression is evaluated against a concatenated list of process arguments. |
| Expression Matching Results | If you use the Select a Process button to connect to a monitored computer and select a process by name, the list of all processes with the selected process name for that computer are shown in this field. When you provide a Regular expression to filter process arguments the processes listed in this field are filtered so that you can preview the filtering by argument. |

Process Template Settings

The following options are available on the Process Template Settings page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Minimum number of instances | The minimum number of running instances of the monitored process. To alert if no instances of the process are running, check the box Generate an alert when the number of process instances is less than the specified value, and input a value of 1. The number of process instances is calculated after filtering by process name and the optional Regular expression to filter process arguments. If the number of running instances is less than the provided value, an alert will be generated. |
| Maximum number of instances | The maximum number of running instances of the monitored process. To alert if more than a specific number of instances of the process are running, check the box Generate an alert when the number of process instances is greater than the specified value, and input the maximum threshold value. The number of process instances is calculated after filtering by process name and the optional Regular expression to filter process arguments. If the number of running instances is greater than the provided value, an alert will be generated. |

Additional Monitoring

In addition to performing the specified monitoring, the UNIX/Linux Process Monitoring template creates a target class that you can use for additional monitors and rules. Any monitor or rule that uses this class as a target runs on any agent where the process is installed.

Creating and Modifying UNIX/Linux Process Monitoring Templates

To create a UNIX/Linux Process Monitoring template

|  |
| --- |
| 1. If you want to monitor a process on a group of computers, determine the targeted group for the monitor by using the following logic:   If you want to discover the process on all UNIX and Linux computers in the management group, you do not have to create a group. You can use the existing group UNIX/Linux Computer Group.   If you only want the process to be discovered on a certain group of computers, then either ensure that an appropriate group exists or create a one by using the procedure in How to Create Groups in Operations Manager 2012.  2. Start the Add Monitoring wizard.  3. On the Select Monitoring Type page, select UNIX/Linux Process Monitoring, and then click Next.  4. On the General Properties page, in the Name and Description boxes, type a name and optional description for this new template.  5. Select a management pack in which to save the monitor or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  6. Click Next.  7. Click the Select a Process button.  8. Click the Browse button and select a computer that has the process installed, and then click OK.  9. In the Processes box, select the process to monitor.  10. Select an Alert Severity.  11. Optionally, provide a Regular expression to filter the matched process list by process arguments, in the field Regular expression to filter process arguments.  12. If you want to monitor the process on a group of computers, do the following:  a. Click the Select a Group button.  b. Select a group that contains the computers with the process, and then click OK.  13. Click Next.  14. Check the appropriate boxes for Generate an alert when the number of process instances is less than the specified value and Generate an alert when the number of process instances is greater than the specified value.  15. Provide values for Minimum number of instances and Maximum number of instances, if appropriate.  16. Click Next.  17. Click Create. |

To modify an existing UNIX/Linux Process Monitoring template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select UNIX/Linux Process Monitoring.  4. In the UNIX/Linux Process Monitoring pane, locate the template to change.  5. Right-click the template, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing UNIX/Linux Process Monitors and Collected Data

To view the state of each monitor

|  |
| --- |
| 1. Select the UNIX/Linux Computers view.  2. In the UNIX/Linux Computers pane, right-click an object, select Open, and then click Health Explorer.  3. Expand the Availability node, and then click the Application/Service Availability Rollup node to view the individual process monitor. |

See Also

[Creating Management Pack Templates](#zba4924f1f1764083aabfa9b46c7d20dc)

Web Application Availability Monitoring Template

The Web Application Availability Monitoring template lets you create availability monitoring tests for one or more web application URLs and run these monitoring tests from internal locations. In addition to state and alert views, you can display the status of these tests in a provided map dashboard and a details dashboard.

Scenarios

Use the Web Application Availability Monitoring template in scenarios where you have to monitor web-based applications from different locations to see if they are working according to certain requirements, which you can specify.

Internal Locations

You might have web applications that must be available at all times at internal locations. Use the Web Application Availability Monitoring template to see which web applications are available from which internal locations.

Monitoring Performed by the Web Application Availability Monitoring Template

By default, the Web Application Availability Monitoring template configures the following monitoring by default. You can modify the monitor in the Change Configuration page of the Web Application Availability Monitoring template.

|  |  |
| --- | --- |
| Monitor description | Default values |
| Web Application Monitor | **** The monitor is enabled by default.  **** Test Frequency: 10 minutes  **** Performance data collection interval: 1 every 10 minutes  **** Test time-out: 45 seconds  **** HTTP status code: 400 (An alert will be generated if the HTTP status code is 400 or greater.)  **** Number of consecutive times a criteria should fail before an alert is generated: 1  **** Generate alerts from each test: enabled  **** Allow redirects: enabled  **** HTTP version: HTTP/1.1  **** HTTP method: GET  **** HTTP headers: accept “/”  **** HTTP headers: accept language of your product  **** HTTP headers: accept encoding GZIP |
| Performance Data Collection | **** Transaction response time: enabled  **** Response time: enabled  **** TCP connect time: enabled  **** Time to first byte: enabled  **** Time to last byte: enabled  **** DNS resolution time: enabled  **** Content size: enabled  **** Content time: enabled  **** Download time: enabled |

Viewing Monitoring Data

All data collected by the Web Application Availability Monitoring template appears in the Web Application Availability Monitoring folder in the Application Monitoring folder in the Monitoring navigation pane. The Application Availability Monitoring folder contains the default views and subfolders that provide Test State, Web Application Status, and alerts related to the tests being monitored. By using the Test State view, you can see the test state of the individual tests. The state of each object matches the state of the targeted object that has the worst health state so that you see the worst state of the monitors that are running. If one or more of the tests are shown with an error while at least one other test is healthy, it could indicate a problem for that particular test location. If all of the components are unhealthy, it could indicate a problem with the web application itself.

Web Application Availability Monitoring folder

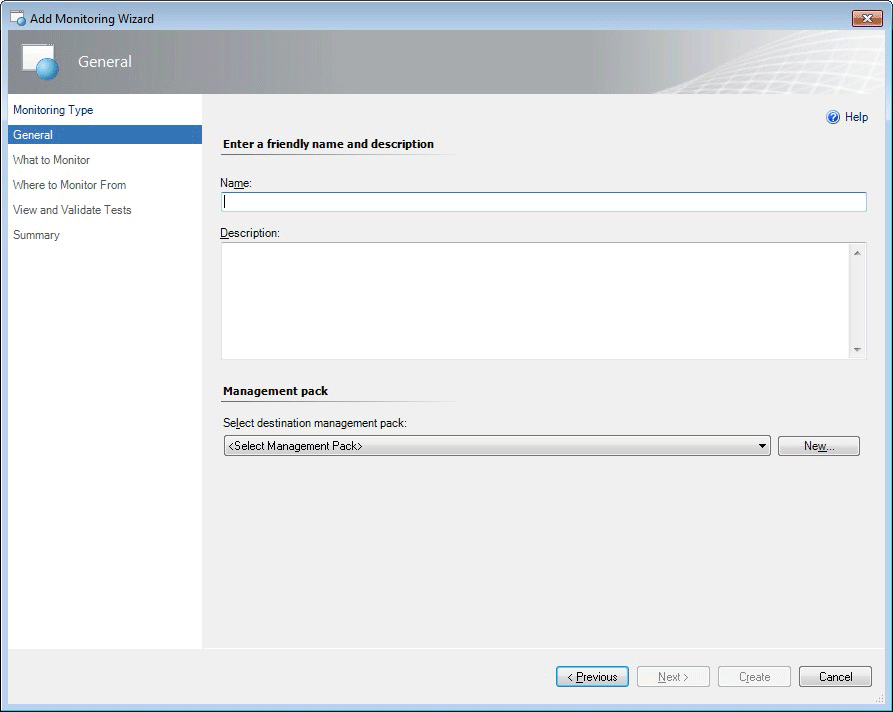


To view the state of the individual monitors, open the Health Explorer for each test.

Wizard Options

When you run the Web Application Availability Monitoring template, you have to provide values for options as listed in the following tables. Each table represents a single page in the wizard.

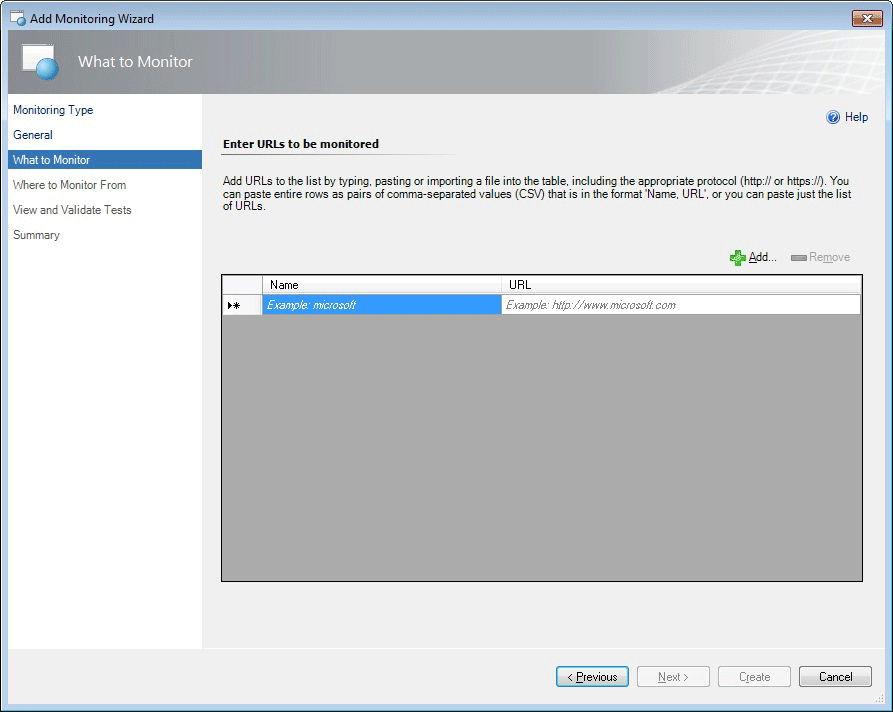
General



The following options are available on the General page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | Enter the friendly name used for the template and test group that you are creating. This name is displayed in the Operations Console in the Web Application status view and is used for the folder under the Web Application Availability Monitoring folder.  Note  After you have given the template a name and saved the template, this name cannot be edited without deleting and re-creating the template. |
| Description | Describe the template. (Optional) |
| Select destination management pack | Select the management pack to store the views and configuration created by the template. Use the same name for your new management pack as the test group so you can easily pair the two names. You can use an existing management pack or create a new management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

What to Monitor

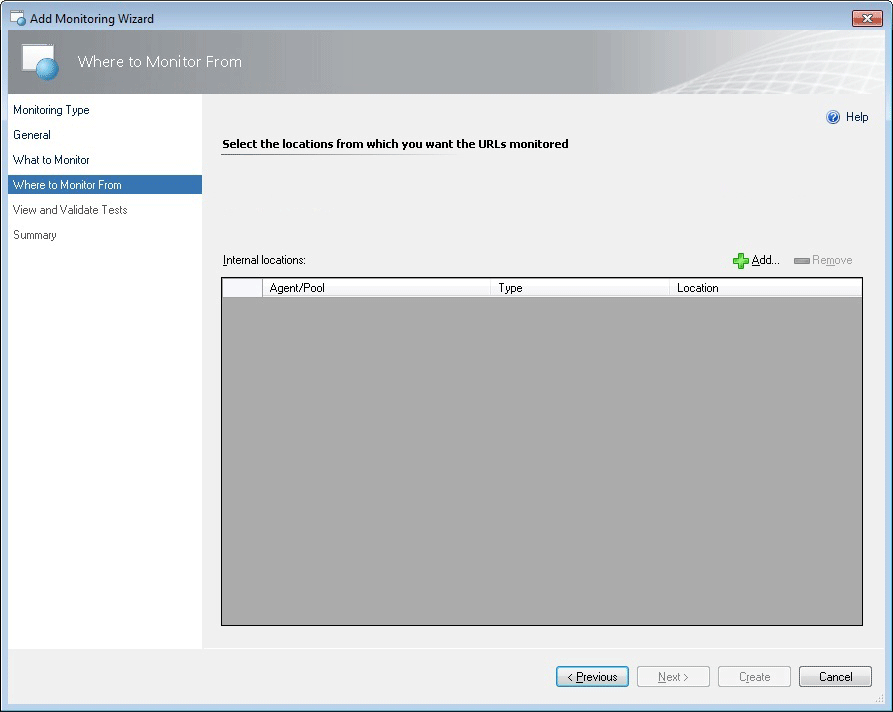


Add URLs to the list by typing, pasting, or importing a file into the table, including the appropriate protocol (http:// or https://). You can paste entire rows as pairs of comma-separated values (CSV) that are in the format ‘Name, URL’, or you can paste just the list of URLs.

The following options are available on the What to Monitor page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | Name of the website you want to monitor. |
| URL | URL of the website you want to monitor in the format: http://www.website.com |
| Add | Add URLs to monitor from an external file. You can paste a list of URLs or rows of a spreadsheet as pairs of comma-separated values that are in the format: Name, URL |

Where to Monitor From

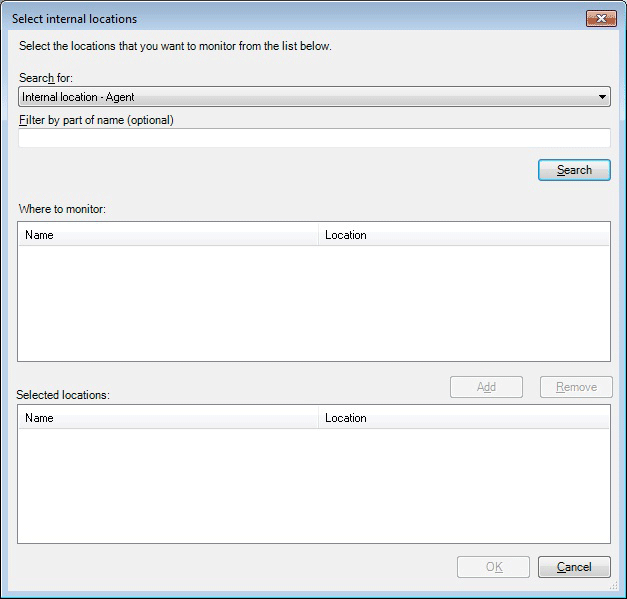


Select the internal locations from which you want the URLs to be monitored.

The following options are available on the Where to Monitor From page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Internal locations | The internal locations you are configuring to monitor from. |
| Add/Remove | Add or remove internal locations you want to monitor from. |

Select Internal Locations

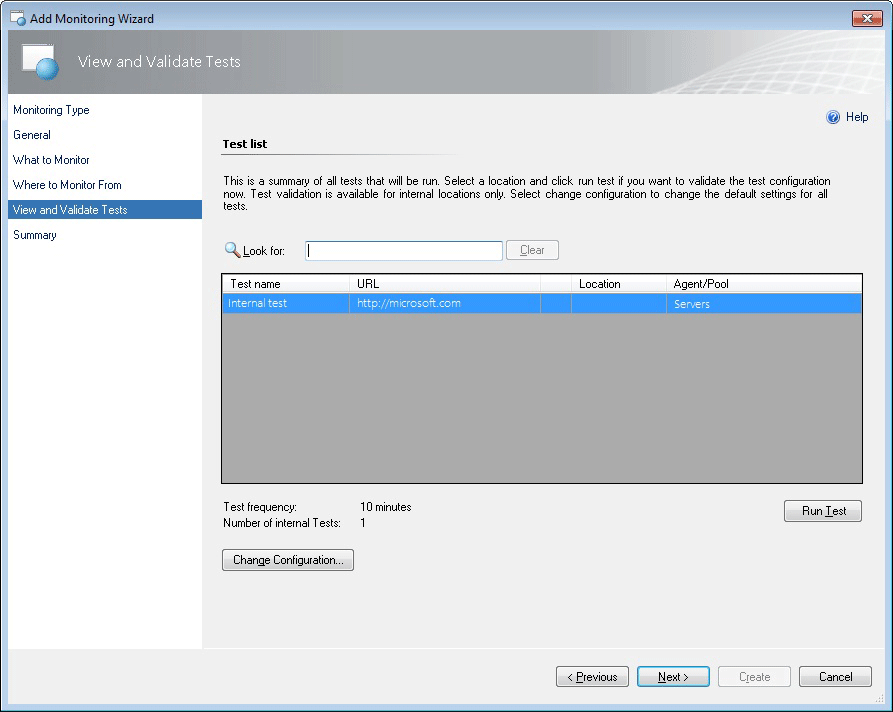


Select the internal locations from which you want to monitor the URLs you specified on the What to Monitor page. Click Add to add internal locations and then search for and select the internal locations that you want to monitor from.

The following options are available on the Select internal locations page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Search for | Option showing the kind of locations you search will look for. You can choose agents or pools. |
| Filter by part of name | Filter your search of internal locations. |
| Search | Search for locations that are available to monitor from. Available locations are displayed in the in the Location area. |
| Where to monitor: Name | List of the internal locations from which you can select to monitor from. |
| Where to monitor: Location |  |
| Add | Add the internal locations you have selected to the Selected locations area. These are the locations you are configuring the wizard to monitor from. |
| Selected locations: Name | These are the internal locations you have chosen to monitor from. |
| Selected locations: Location | List of the locations you have chosen to monitor from. |

View and Validate Tests



This is a summary of all tests that will be run. Select an internal location and click Run Test to validate the test configuration. Select Change configuration to change the default settings for all tests in this template.

The following options are available on the View and Validate Tests page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Look for | Search for and returns results for items in the list of test names, URLs, Locations, and Agent/Pools. Use this to find specific tests or sets of tests that you want to validate. |
| Test Name | Name of a test. |
| URL | URL for a specific test. |
| Agent/Pool | The Agent or Pool location for your internal URL tests. |
| Run Test | Run a validation test for internal tests that are selected. |
| Change Configuration | Open the Change Configuration page where you can change the settings for all tests in the template you are authoring. |

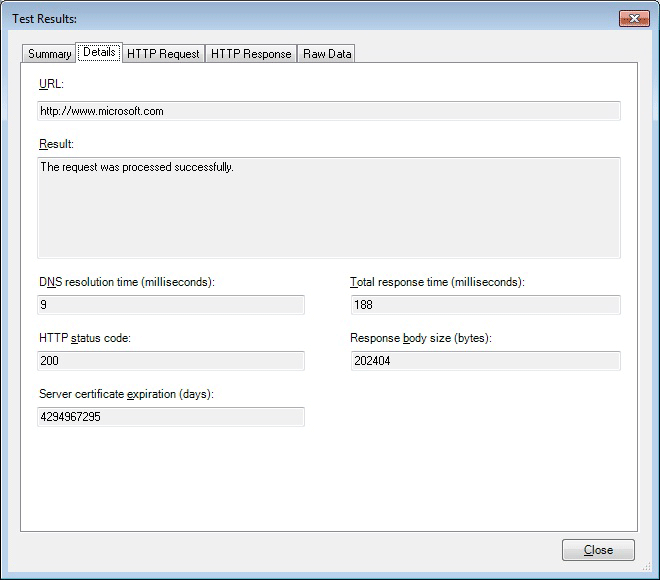
Test Results: Summary Tab



The following options are available on the Test Results Summary tab of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Summary tab | Confirms if the test request was correctly processed and shows the URL and Location used in the test. Additionally. The specific tests and results are shown: Status code, DNS resolution time, and Total response time. |

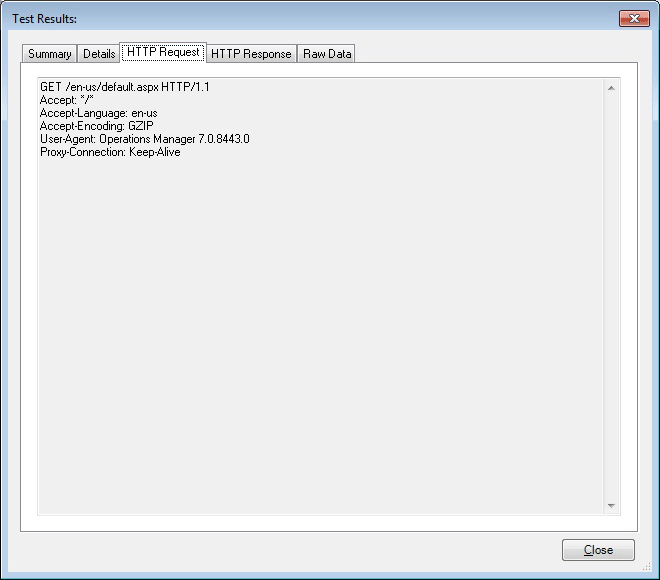
Test Results: Details Tab



The following options are available on the Test Results Details tab of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Details tab: URL | See detailed information about the test. Displays which URL was tested. |
| Details tab: Result | Displays whether the test request was processed successfully or not. |
| Details tab: DNS resolution time (milliseconds) | Displays the DNS resolution time which checks that website performs as you expected it to. What’s the IP address of the URL you are. Time it takes for DNS to get the IP address for the website. |
| Details tab: Total response time (milliseconds) | Displays the Total response time from same as transaction time performance counter. |
| Details tab: HTTP status code | Displays the HTTP status code when you ping a website, you get a status code. |
| Details tab: Response body size (bytes) | Displays the Response body size of the HTTP response information. |
| Details tab: Server certificate expiration (days) | Displays the certificate expiration of the date when the site expired. Website can have expired certificates. |

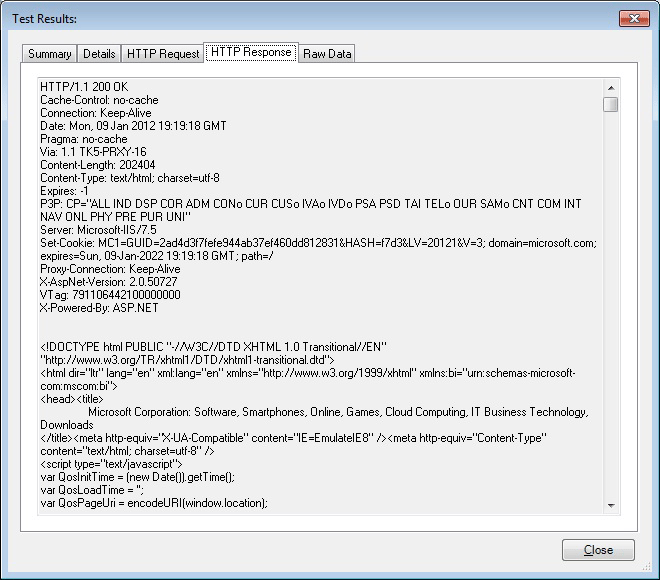
Test Results: HTTP Request Tab



The following options are available on the Test Results HTTP Request tab of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| HTTP Request tab | Displays details about the HTTP request of the test what is sent to the website. |

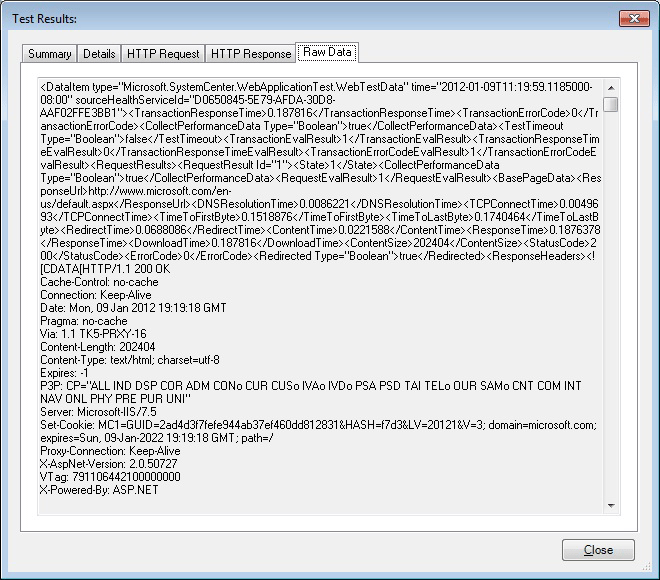
Test Results: HTTP Response Tab



The following options are available on the Test Results HTTP Response tab of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| What is shown on this tab | Displays details about the HTTP Response for the test comes back from website. |

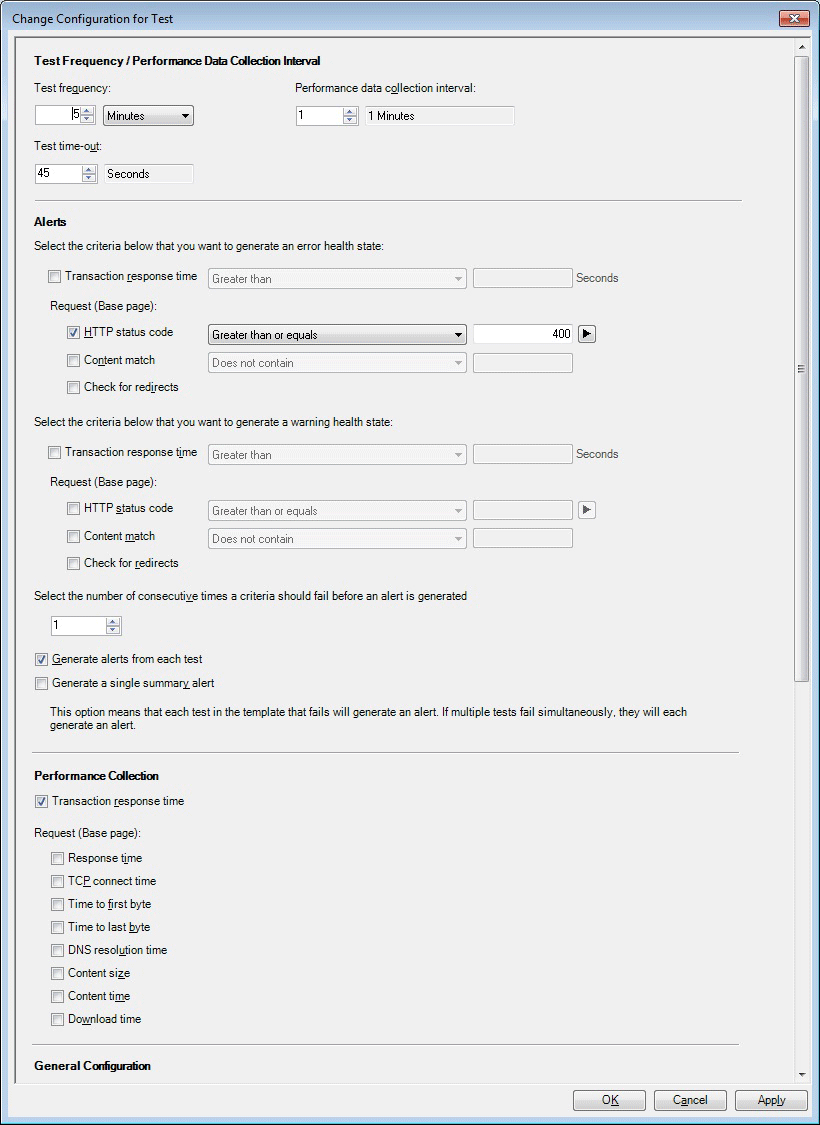
Test Results: Raw Data Tab

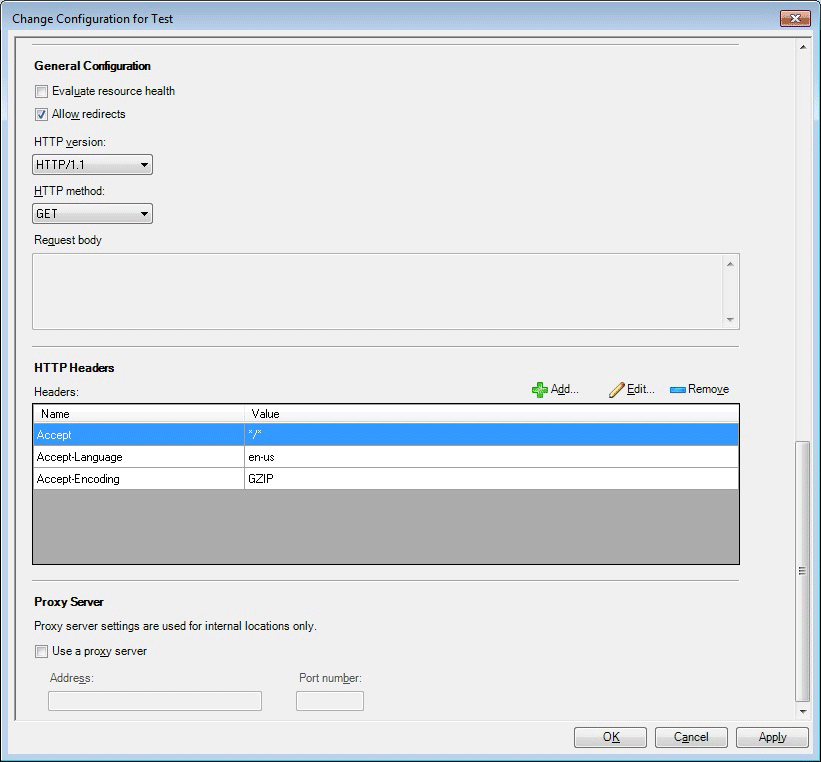


The following options are available on the Test Results Raw Data tab of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| What is shown on this tab | Displays all of the data unformatted that we get back from the site. If there’s a problem with the website, this information might help you figure out what might be wrong with the website. |

Change Configuration for Test Set





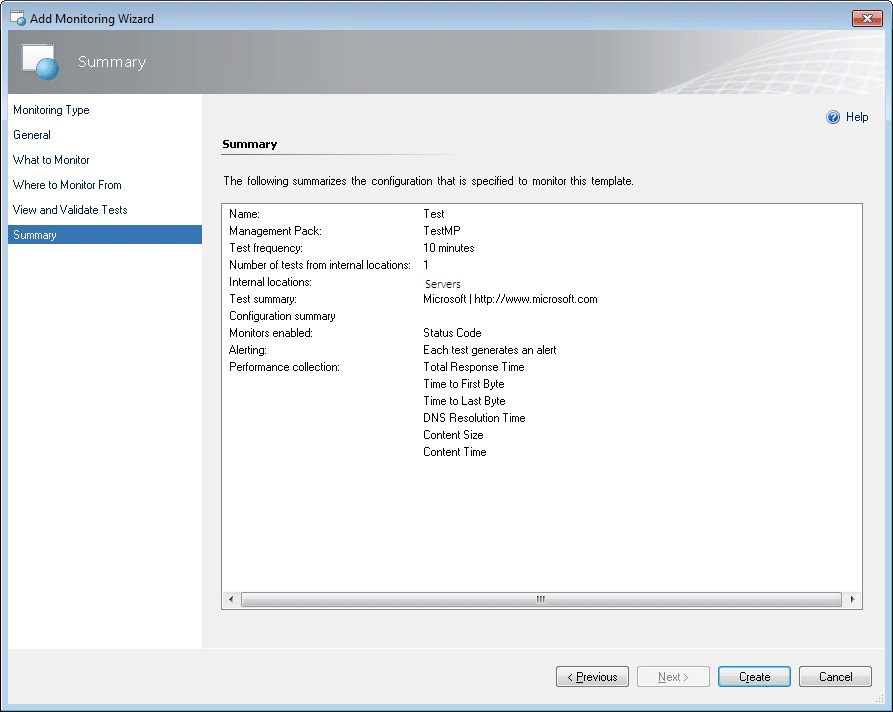
The following options are available on the Change Configuration for Test Set page of the wizard.

Important

Settings on this page apply to all tests in the template.

|  |  |
| --- | --- |
| Option | Description |
| Test Frequency/Performance Data Collection Interval: Test frequency | Enter the how often you want to run each test. |
| Test Frequency/Performance Data Collection Interval: Performance data collection interval | Enter the frequency with which you want to collect performance data. This specifies whether you want to collect performance data every interval or not. For example, if the interval is 10 minutes and the collection interval is set to 2, this means that performance data will be collected every other interval, or once every 20 minutes. |
| Test Frequency/Performance Data Collection Interval: Test time-out | Enter how long you want the test to keep a request active until the test times out and cancels. |
| Alerts: Criteria for error health state: Transaction response time | Specify if transaction response time is a factor that should or should not generate an error health state. If it is specified to generate an error health state, set the threshold in seconds that a transaction must exceed before it generates an error health state. |
| Alerts: Criteria for error health state: Request (Base page): HTTP status code | Specify if the HTTP status code is a factor that should or should not generate an error health state. If it is specified to generate an error health state, set the HTTP status code to the number for which you want it to generate an error health state. |
| Alerts: Criteria for error health state: Request (Base page): Content match | Specify if any content matches should or should not generate an error health state. If it is specified to generate an error health state, specify the content you wish to match. |
| Alerts: Criteria for error health state: Request (Base page): Check for redirects | Specify if the presence of redirects should or should not generate an error health state. |
| Alerts: Criteria for warning health state: Transaction response time | Specify if the transaction response time is a factor that should or should not generate a warning health state. If it is specified to generate warning health state, set the threshold in seconds that a transaction must exceed before it generates a warning health state. |
| Alerts: Criteria for warning health state: Request (Base page): HTTP status code | Specify if the HTTP status code should or should not generate a warning health state. If it is specified to generate warning health state, set the HTTP status code to the number for which you want it to generate a warning health state. |
| Alerts: Criteria for warning health state: Request (Base page): Content match | Specify if any content matches should or should not generate a warning health state. If it is specified to generate a warning health state, specify the content you wish to match. |
| Alerts: Criteria for warning health state: Request (Base page): check for redirects | Specify if the presence of redirects should or should not generate a warning health state. |
| Alerts: Number of consecutive time a criteria should fail before an alert is generated | Specify the number of consecutive times selected criteria in the Alerts section list should fail before an alert is generated. |
| Alerts: Generate alerts from each test | Select to receive an alert for each URL test for an application. |
| Alerts: Generate a single summary alert | Select to receive a summary alert for an application, rather than choosing to receive an alert for each URL test for an application. This is helpful if you are monitoring a vertical website or an application because this will reduce the number of alerts you receive and keep the focus of your alerts the overall state of the application.  You can further reduce alerts by raising the threshold for how many failures you want to have before receiving an alert. Together, these two approaches will focus your alerts on what is most important to you: How well the application is running, given the performance you require. |
| Performance Collection: Transaction response time | Cumulative response time: DNS\_RESOLUTION\_TIME + TCP\_CONNECT\_TIME + TIME\_TO\_LAST\_BYTE |
| Performance Collection: Request (Base page): Response time | Processing time for the request, such as opening a browser and waiting for all resources to load. |
| Performance Collection: Request (Base page): TCP connect time | Time taken to establish a TCP connection to the target server and receive the initial greeting from the service. |
| Performance Collection: Request (Base page): Time to first byte | Time take since the TCP connection is established till the first byte of response is received. |
| Performance Collection: Request (Base page): Time to last byte | Time from when TCP connection is established until the last byte of response is completely received. |
| Performance Collection: Request (Base page): DNS resolution time | Time taken to resolve the URL domain name to the IP address. |
| Performance Collection: Request (Base page): Content size | Size of the response body received. |
| Performance Collection: Request (Base page): Content time | Base page download time (base page only). |
| Performance Collection: Request (Base page): Download time | Processing time for the request, such as opening a browser and waiting for all resources to load. |
| General Configuration: Evaluate resource health | Specify whether to evaluate the health of the entire resource. |
| General Configuration: Allow redirects | Specify if redirects can be allowed and not cause an error or warning state. |
| General Configuration: HTTP version | Specify the HTTP version being tested. |
| General Configuration: HTTP method | Specify the HTTP method. |
| General Configuration: Request body |  |
| HTTP Headers: Headers column | Specify which headers can be accepted. |
| HTTP Headers: Value column | Specify the value in the header that can be accepted. |
| HTTP Headers: Add | Add header names and values that can be accepted. |
| HTTP Headers: Edit | Opens the HTTP Header Properties page where you can change the Name or Value of the selected HTTP headers. |
| HTTP Headers: Remove | Removes selected header from the accepted list. |
| Proxy Server: Use a proxy server | Specify whether to use a proxy server. |
| Proxy Server: Address | Specify the address of the proxy server. |
| Proxy Server: Port number | Specify the port number. |

Summary



The Summary page of the wizard lists the settings you have configured for the Web Application Availability Monitoring template. If you want to change any of these settings, click Previous or the template page until you reach the page with the settings that you want to change.

Creating and Modifying Web Application Availability Monitoring Templates

For the procedure to run the .NET Application Performance Monitoring wizard, see [How to Configure Web Application Availability Monitoring](#z340c7bbcca874bf28fe848b8a141aac4)

To modify an existing Web Application Availability Monitoring template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials in the management group.  2. Click the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select Web Application Availability Monitoring.  4. In the Web Application Availability Monitoring pane, locate the template you want to change.  5. Right-click the test group that you want to modify, and then select Properties.  6. Using the tabs to navigate the pages of settings, make the desired changes, such as reconfiguring criteria for tests in this group, and then click OK. |

Viewing Web Application Availability Monitoring Monitors and Collected Data

After you configure monitoring for an application, these three views will help you get started with the monitoring experience.

To view all Web Application Availability Monitoring monitored applications

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand Web Application Availability Monitoring, and then click Web Application Status. |

To view the state of each monitor

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand Web Application Availability Monitoring, and then click Test State.  4. In the Test State view, right-click an object. Select Open, and then click Health Explorer. |

To view the performance collected for an application component

|  |
| --- |
| 1. Open the Operations console.  2. Click the Monitoring workspace.  3. In the Monitoring navigation pane, expand Application Monitoring, expand Web Application Availability Monitoring, and then click Web Application Status.  4. In the Test State pane, right-click an object. Select Open, and then click Performance View.  5. In the Legend pane, select the counters that you want to view.  6. Use options in the Actions pane to modify the Performance view. |

See Also

[How to Configure Web Application Availability Monitoring](#z340c7bbcca874bf28fe848b8a141aac4)

[Monitoring Web Application Availability Tests and Alerts](#zd786ccb26b4941219f1b21414f29dc95)

[Dashboard Views for Web Application Availability Monitoring](#z8605ec589fef47159f74a195eefb82c3)

[Reporting for Web Application Availability Monitoring](#z7a15fc3cc0964ea1b7682df47289a44a)

How to Configure Web Application Availability Monitoring

The Web Application Availability Monitoring template in System Center 2012 – Operations Manager lets you monitor web applications from internal locations.

Key Concepts and Strategies

Here are some key concepts and strategies to keep in mind as you use the template to configure tests:

 Web Application Availability Monitoring can monitor applications as well as individual URLs. Your tests can be authored to optimize for application monitoring or vertical application feature monitoring. For example, a website shopping experience is a vertical application feature. Alternatively, you can monitor horizontal application features, such as all home pages on a site.

 Give your tests friendly names you can easily recognize when you monitor.

 Group URLs by Web Application and you will see one entry in the monitoring view for each application. This lets you see the state of each application template run at a glance and helps isolate problems related to servers or networks.

 Web Applications State is a defined as the worst of rollup of individual tests. You can use overrides to change what it displays.

 Validate each test configuration before exiting the Add Monitoring Wizard so you can check that the response you get is the one you expect for internal watcher nodes.

 If you are monitoring a vertical website or an application, you can focus alerts on application state by setting up a summary alert for an application, rather than choosing to receive an alert for each URL test for an application. This will reduce the number of alerts you receive and keep the focus of your alerts the overall state of the application. You can further reduce alerts by set the threshold for how many failures you want to have before receiving an alert. Together, these two approaches will focus your alerts on what is most important to you: how well the application is running.

How to Configure Monitoring

To configure Web Application Availability Monitoring

|  |
| --- |
| 1. To configure internal URL tests, use the Web Application Availability Monitoring template. This template allows you to create availability monitoring tests for one or more Web Application URLs and run them from internal locations and will walk you through selecting the URLs you want to monitor and where you want to monitor them from.  To open the Web Application Availability Monitoring template, in the Operations Manager console, in the navigation pane, click the Authoring button, click Management Pack Templates, click Web Application Availability Monitoring, and then, in the tasks pane, click the Add Monitoring Wizard. Click Web Application Availability Monitoring, and then click Next.  Location of Web Application Availability Monitoring    2. On the General page, enter a friendly name and description for your test and the management pack, which stores settings that are specific to this version of the template. Click Next  Tip  It is important to give your tests names you can easily recognize as this will help you identify them later when you begin monitoring.  3. On the What to Monitor page, enter the URLs you want to monitor. You can add URLs to the list by typing, pasting, or importing a file (such as a spreadsheet) into the table, including the appropriate protocol (http:// or https://). You can paste entire rows as pairs of comma-separated values (CSV) that are in the format “Name, URL”, or you can paste just the list of URLs.   To type or paste a URL, click in the Name and URL boxes, respectively, and follow the examples on the page to enter the information.   To import names and URLs from another source, click Add and follow the instructions to browse and select to the file you want to import.  Tip  Group URLs by web application. Example: Microsoft website would be the name of many URLs to test from various locations. This keeps the application, Microsoft website, the focus of the monitoring rather than the many URLs you are testing for this application. This strategy takes advantage of the dashboard monitoring views, which show the Web application state by application name and state so you can easily see the overall health of your monitored application and separate problems related to external factors, such as Internet or network problems, from application or service problems.  4. On the Where to Monitor From page, select the locations from which you want the URLs monitored.  5. To monitor from internal locations, use the dropdown menu to select to run tests from a server agent or a resource pool. Above the Internal locations field, click Add and select the internal locations (server names) you want to monitor from and add those to your list of Selected locations. Click OK  Note  The Select internal locations page verifies the version of health service on the agents and only Operations Manager agents are shown. If the version is not verifiable or not supported, the agents will not be displayed.  6. The View and Validate Tests page shows a summary of all internal tests that will be run. You can only validate a configuration of internal tests. To validate an internal test configuration, select a test and click Run Test. If you want to change the default settings for the tests, performance data, and alerts for all tests (external and internal) created in this template, click Change Configuration. For more information, see the “Configuring alerts and performance data collection” in this document. When you have validated your tests, click Next to see the Test Results page of the test configuration you created with this template.  Tip  Validate internal test configurations before exiting the wizard so you can make sure that the response you get is the one you expect.  7. On the Test Results page, the Summary tab shows if your test succeeded and whether any of your error criteria failed. To see more details, click through the Details, HTTP Request, and HTTP Response tabs. Click Close to return to the View and Validate Tests page where you can either choose to reconfigure your tests, or, if the test results look good, click Create to initiate the live testing. |

To configure alerts and performance data collection

|  |
| --- |
| 1. Configure alerts and performance during test configuration. To configure alerts, on the View and Validate Tests page, click Change Configuration. Here you can set the criteria that are most important for you to monitor. These are the URL monitoring performance counters you can set and a description of what each tracks. Most of these are collected by default. For detailed information about the authoring template, see [Web Application Availability Monitoring Template](#z62d3b2b3f4ff4fe180553078179c1b1e)  Base page (the page served by the request URL)   Content size: Size of the response body received.   DNS resolution time: Time taken to resolve the URL domain name to the IP address.   Content time: Base page download time (base page only).   Download time (cumulative): Processing time for the request, such as opening a browser and waiting for all resources to load.   TCP connect time: Time taken to establish a TCP connection to the target server and receive the initial greeting from the service.   Time to first byte: Time take since the TCP connection is established till the first byte of response is received.   Time to last byte (cumulative): Time from when TCP connection is established until the last byte of response is completely received.   Response time (cumulative): DNS\_RESOLUTION\_TIME + TCP\_CONNECT\_TIME + TIME\_TO\_LAST\_BYTE.  Total data   Total transaction time: Same a download time for the single URL case. In multi-step (transaction) case, it’s the sum of download times for all requests in the transaction (cumulative).  2. In the Test Frequency/Performance Data Collection Interval section, you can set your tests’ frequency, how often you want performance data collected, and when the test should time out.  3. In the Alerts section, select the criteria that you want to use to generate an error health state, a warning health state, and the number of consecutive times criteria should fail before an alert is generated. When you monitor your alerts, here is how they will appear based on your alert configuration:   Healthy state=Green   Error state=Yellow   Warning state=Red  Tip  If you are monitoring a vertical website or an application, you can focus alerts on application state by setting up a summary alert for an application, rather than choosing to receive an alert for each URL test for an application. To do this, select the Generate a single summary alert checkbox. This will reduce the number of alerts you receive and keep the focus of your alerts the overall state of the application. You can further reduce alerts by raising the threshold for how many failures you want to have before receiving an alert. Together, these two approaches will focus your alerts on what is most important to you: How well the application is running, given the performance you require.  4. You also can change settings for Http Headers, designate a Proxy Server (used for internal locations only), and change other general settings. When you have finished configuring the settings for your tests, click OK to return to the View and Validate Tests page. |

Monitoring Web Application Availability Tests and Alerts

To view overall web application health

|  |
| --- |
| 1. To get a view of the general health of each application you are monitoring, in the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  2. Click Web Application Availability Monitoring, and then click Web Applications State. This is the best view to see the overall health of each application.  The alert state for each application is displayed according to the alert configurations used when you set up the tests:   Green=Healthy   Yellow=Error   Red=Warning  3. To see details about a particular application, double-click the test to open the Health Explorer.  The Health Explorer allows you to see the health states for each criterion, so you can pinpoint what caused the application to show an error or warning. |

To view alerts and alert details

|  |
| --- |
| 1. To see active alerts for the applications you are monitoring, in the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  2. Click Web Application Availability Monitoring, and then click Active Alerts. If there are alerts, you will see them listed in the Active Alerts pane. As with the application health, the alert state is displayed in the color according to your alert configurations.  3. To see alert details, click the alert to see the Alert Details pane.  4. To see more details about an alert, double-click the alert to open the Alert Properties page.  On the Alert Properties page General tab, you can reassign the alert, see the alert description, and change the alert status to New or, if you have resolved the issue, Closed. Use the Previous and Next buttons to scroll through the alerts for easy review. When you are finished viewing or changing properties, click OK to save your changes. |

To view data for individual URLs and locations

|  |
| --- |
| 1. To get a view of each test you configured in the Web Application Availability Monitoring template, in the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  2. Click Web Application Availability Monitoring, and then click Test State. If there are alerts, you will see them listed in the Active Alerts pane. As with the application health, the alert state is displayed in the color according to your alert configurations.  Note  A test is one URL and one location.  3. To see the details of a particular test, double-click the test to open the Health Explorer.  The Health Explorer allows you to see the health states for each criterion, so you can see in detail what caused the test to show an error or warning. |

Dashboard Views for Web Application Availability Monitoring

Dashboards show your test results so you can more effectively see trends and isolate problems for certain tests, locations, and applications.

Viewing the Dashboards

To view the Web Application Availability Monitoring dashboards for internal URL tests, you need to use Operations Manager script.

To view the Summary and Details dashboards

|  |
| --- |
| 1. From the Start menu, open the Operations Manager Shell.  2. Use New-SCOMLocation, Get-SCOMAgent, and Set-SCOMLocation to add a location, get an Operations Manager agent, and then associate the location with the agent. Here is an example using Seattle, WA as a location:  PS C:\> $Location = New-SCOMLocation -DisplayName “Seattle, WA” -Latitude 47.6063889 -Longitude -122.330833  PS C:\> $Agent = Get-SCOMAgent -Name “Server01.Contoso.com”  PS C:\> Set-SCOMLocation -Location $Location -Agent $Agent  For more information about adding a location, see [New-SCOMLocation](http://go.microsoft.com/fwlink/?LinkId=235473). For more information about getting an Operations Manager agent, see [Get-SCOMAgent](http://go.microsoft.com/fwlink/?LinkID=187686). For more information about associating a location with an agent, see [Set-SCOMLocation](http://go.microsoft.com/fwlink/?LinkId=235479).  3. After you run the commands, in the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  4. Expand Web Application Availability Monitoring, click Active Alerts, Test State, or Web Application Status to display alerts or status for the category.  5. From any of these views (alert, test state or web application), click to highlight the application you want to see a summary of and follow the procedure to view the dashboards in this topic. |

The Summary Dashboard

If you want to check to see if an application is available, the Summary Dashboard is a helpful view. It displays a world map, the locations you are monitoring from, and the rollup test status from each location. You can then click the location or select several locations to compare.

To check the overall status of a web application using the Summary Dashboard

|  |
| --- |
| 1. In the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  2. Expand Web Application Availability Monitoring, and then click Active Alerts, Test State, or Web Application Status to display alerts or status for the category.   Active Alerts displays all alerts for all web applications (templates) or tests.   Test State displays the state of each test, which is based on how you configured the tests. You can click State, Display Name, and Context column headings to sort the test state list accordingly. For example, if you sort by Context, this groups the list by web applications, so you can see the state of the tests for each web application.   Web Application Status displays the worst rollup of tests for each web application. This is a quick way to check whether an application is having any issues.  3. From any of these views (alert, test state or web application), click to highlight an entry for the application you want to see a summary of. Then, in the Tasks pane, in the Navigation section, click Summary Dashboard. This displays a dashboard for the entire application—a world map that shows all locations the application is being tested from and the worst rollup status of all of the tests from each location. For example, red means that at least one test from that location has failed and green means that no tests have failed.  4. To see the test status for all tests from a given location, click that location and the test status displays below.  5. If you want to investigate a particular test, or group of tests, click to highlight the tests you want to investigate and in the Tasks pane, in the Navigation section, click Detailed Dashboard – List. This opens detailed views into the item you clicked. For more information, see the details dashboard section below. |

The Details Dashboard

If you want to investigate a particular test or alert, use the Detailed Dashboard – List. For each web application, you choose the location and which tests in that location you want to investigate. Six key metrics are shown, which you can use to pinpoint and isolate issues and compare the performance of pages from your web applications or compare your pages to competitors’ pages.

To pinpoint problems using the Details Dashboard

|  |
| --- |
| 1. Begin the same way you opened the Summary Dashboard. In the Operations Manager console, in the navigation pane, click the Monitoring button, and then click Application Monitoring.  2. Expand Web Application Availability Monitoring, and then click Active Alerts, Test State, or Web Application Status to display alerts or status for the category.  3. From any of these views (alert, test state or web application), click to highlight the application you want to see a summary of, and in the Tasks pane, in the Navigation section, click Detailed Dashboard - List.  Tip  If you are already viewing the Summary Dashboard, you can go directly to the Detailed Dashboard by clicking to highlight a test or group of tests and then in the Tasks pane, in the Navigation section, clicking Detailed Dashboard.  4. In the section with the name of your application, for example Bing template select the locations you want to see more details about.  5. In the Test Status section, the check boxes act as a legend for the tests you want to investigate in the performance graphs. For example, you can select the same page being tested from different locations to see how the pages are performing for the different locations. Selection (blue highlighting) determines which tasks are available and can be run. |

Health Explorer

As its name implies, Health Explorer allows you to see more details about the health status of a web application availability test running against a URL from a particular location. Health Explorer shows you when a test changed state from, for example, healthy to unhealthy.

To view the context of a problem using Health Explorer

|  |
| --- |
| 1. To open Health Explorer for a particular test, highlight an alert, test state, or web application status item, and in the Tasks pane, click Health Explorer. If you are already in a dashboard view, you can also open Health Explorer in the Summary Dashboard and Detailed Dashboard by right-clicking an entry, and then clicking Health Explorer.  2. In Health Explorer, in the pane named with your test, click to highlight an item you want to investigate.  3. Click the State Change Events tab to see details on when a state changed from healthy to unhealthy.  4. In the State Change Events Details pane, you can see the error details that caused the health state change of your test.  Tip  It is a good idea to check Health Explorer frequently because it shows you details about the context and sequence of status changes and errors. |

Reporting for Web Application Availability Monitoring

Web Application Availability Monitoring introduces to two new report tasks: Test Availability and Test Performance. The Test availability reporting shows measures reflecting how available the web application was over time. The Test Performance report shows selected objects and performance counter values over time to relate how well a web application has performed. These two reports directly reflect the web application monitoring for Web Application Availability is integrated into the Operations Manager console, you can run these two new reports just as you would run the other standard Operations Manager reports.

To run the Test Availability report

|  |
| --- |
| 1. You can run the Test Availability report from several locations:   In the Operations Manager console, in the navigation pane, click the Monitoring button, click Application Monitoring, click Web Application Availability Monitoring, and then click Active Alerts, Test State, or Web Application Status to display alerts or status for the category. Highlight one or more web applications, alerts, or test states, and then, in the Tasks pane, in the Report Tasks section, click Test Availability.   If you want to add more objects, click Add Object, and in the Add Object page, filter your search or just click Search to see all tests in for the web application that you have selected. Click the available items you want to include in your report and click Add. Click OK.   In the Summary Dashboard view, click to highlight an item in the Test Status, and then run the report task from the Tasks pane.   In the Details Dashboard, click to highlight an item, and then run the report task from the Tasks pane.  2. In the Aggregation section of the Test Availability report configuration page, select how often you want information aggregated and a time and date range.  Note  The data warehouse only aggregates data hourly for 100 days. After 100 days, data are aggregated daily.  3. Select the aspects, such as Unplanned Maintenance, that you would like to include in the report, and then click Run to generate the report.  4. In the report, click a plus sign to see details. To see a graph of the data, click the Availability Tracker link. |

To run the Test Performance report

|  |
| --- |
| 1. You can run the Test Performance report from several locations:   In the Operations Manager console, in the navigation pane, click the Monitoring button, click Application Monitoring, click Web Application Availability Monitoring, and then click Active Alerts, Test State, or Web Application Status to display alerts or status for the category. Highlight one or more web applications, alerts, or test states, and then, in the Tasks pane, in the Report Tasks section, click Test Performance.   If you want to add more objects, click Add Object, and in the Add Object page, filter your search or just click Search to see all tests in for the web application that you have selected. Click the available items you want to include in your report and click Add. Click OK.   In the Summary Dashboard view, click to highlight an item in the Test Status, and then run the report task from the Tasks pane.   In the Details Dashboard, click to highlight an item, and then run the report task from the Tasks pane.  2. In the Aggregation section of the Test Performance report configuration page, select how often you want information aggregated and a time and date range.  Note  The data warehouse only aggregates data hourly for 100 days. After 100 days, data are aggregated daily.  3. To generate the report, click Run.  4. In the report, click a plus sign to see details. |

Web Application Transaction Monitoring Template

The Web Application Transaction Monitoring template lets you test a website or web-based application by sending requests over HTTP, validating their response, and measuring their performance. This can be a simple test to determine if the website is responding, or it can be a complex set of requests to simulate a user who is performing such actions as logging on to the site and browsing through a set of pages.

The HTTP requests are sent from one or more [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1). The website that is being monitored can reside on any computer whether it has an agent for Operations Manager installed. It can be an external website, but it must be accessible from the watcher nodes. Each watcher node must have an Operations Manager agent installed.

Scenarios

Use the Web Application Transition Monitoring template for monitoring the availability and performance of any website or web-based application to test both general availability and functionality. For internal websites, you can use watcher nodes in different network segments to ensure that the site is available to each segment.

In addition to general availability, you can check the functionality of the website by testing different pages and features. For example, you could check a logon process by performing a test logon with a test user account every few minutes. You could test the functionality of a search page by performing a sample search after the test user account is logged on. You can then analyze the HTML that is returned from these pages to verify whether the page functioned as expected. In addition to testing this functionality, you can analyze the time it takes to fill the request to measure the performance.

Web Application Transaction Monitoring Template Topics

The Web Application Transaction Monitoring is more complex than the other management pack templates and supports a variety of monitoring scenarios. The following topics provide details on the different tools and procedures that you can use for different scenarios.

 [How to Create a Single URL Web Application Monitor](#z184a636344d54c54bdf8da324cbcbd7f)

This procedure explains how to run the Web Application Transaction Monitoring wizard to create a simple Web Application Transaction Monitoring template that monitors a single URL. You can either use this simple monitor with no modification or insert additional requests once it is completed.

 [How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e)

 This procedure explains how to use the Web Recorder to record a browser session with multiple requests. You can either use this session with no modification or edit the application and request settings once it is completed.

 [How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e)

This topic includes procedures for editing a web application and its contained requests and for creating additional requests in an existing template.

 [How to Replace Parameters in a URL Request](#zbdc68ab7419e4696a75627f5862b02a3)

 This topic includes a method to retrieve information from one request and use it in a subsequent request.

How to Create a Single URL Web Application Monitor

The most basic Web Application Transaction Monitoring template includes a single request. This may be sufficient for testing the general availability of a website, or as the basis for creating a more complex monitoring scenario by adding requests either manually or by capturing them in a browser session.

To create a single URL web application transaction monitoring template

|  |
| --- |
| 1. Start the Operations console with an account that has Author user rights in the management group.  2. Open the Authoring pane.  3. Right-click Management Pack Templates, and then select Add Monitoring Wizard.  4. On the Select Monitoring Type page, select Web Application Transaction Monitoring, and then click Next.  5. On the General Properties page, do the following:  a. In the Name and Description boxes, type a name and an optional description.  b. Select a management pack in which to save the monitor, or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  6. On the Web Address page, do the following:  a. In the URL box, specify that the request uses http or https.  b. Type the URL to connect to.  c. Click the Test button to perform a test connection to the URL. If the connection succeeds, click Details to inspect the response time and other details of the connection. This information can be valuable if you decide to manually edit the request after the template is created.  Note  The test is performed from the workstation that you are using to run the wizard. If this workstation cannot access the website, this test fails. When the template is completed, the test is run from the watcher nodes that you specify.  d. Click Next.  7. On the Watcher Node page, do the following:  a. Select one or more [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1) to run the web requests.  b. Specify the frequency to run the web request in the Run this query every box.  c. Click Next.  8. On the Summary page, review the summary of the monitor, and then click Create. |

See Also

[How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e)

[How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e)

How to Capture Web Application Recording

The fastest and easiest way to create a Web Application Transaction Monitoring template with multiple requests is to record a session from Internet Explorer. Using the Web Recorder, you can interactively record a sequence of actions that are stored in the request sequence with default settings. After the session has been recorded, you can manually edit the individual requests if any of them require unique configuration.

You can run the Web Recorder to create a Web Application Transaction Monitoring template, or you can run it from an existing template to add additional requests.

To record a new web application

|  |
| --- |
| 1. Start the Operations console with an account that has Author user rights in the management group.  2. Open the Authoring pane, expand Management Pack Templates, and then click Web Application Transaction Monitoring.  3. In the Actions pane, click Record a browser session.  4. In the Web Application Editor dialog box, do the following:  a. Type a name and optionally a description of the recording. The name appears in the Monitoring pane.  b. Select a management pack in which you want to save your Web Application object, and then click OK. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  5. Click Start capture.  Follow the steps in [To use the Web Recorder](#z61). |

To add a web recording to an existing browser session

|  |
| --- |
| 1. Start the Operations console with an account that has Author user rights in the management group.  2. Open the Authoring pane, expand Management Pack Templates, and then click Web Application Transaction Monitoring.  3. Select the template that you want to edit, and then click Edit Web Application settings.  4. Select the location in the browser session to include the recorded requests.  5. Click Start capture in the Actions pane.  Follow the steps in [To use the Web Recorder](#z61). |

To use the Web Recorder

|  |
| --- |
| 1. When you start the capture, Internet Explorer opens with the Web Recorder in the navigation pane. If you do not see the Web Recorder, see [Troubleshooting Web Capture](#z4355df5cdb7e480594dd3e5b5281389b).  2. In the browser window, follow the actions that you want to be monitored. For example, you might click some links or add a product to a shopping cart. As you perform actions, they are recorded in the Web Recorder pane.  3. When you have completed the recording, click Stop in the Web Recorder pane. Internet Explorer closes, and the actions you performed are added to the Web Application Editor.  4. Optionally, click Run Test in the Actions pane to immediately run the recorded actions and view the results. At this point, you might encounter the following errors:   If the web application requires authentication, running a test of the web application might fail. While running the test, credentials that have been configured for this web application are not used. If the site you are testing does not explicitly require authentication, the test might still succeed. In the Actions pane, under Web Application, you can click Configure Settings for any website to select authentication settings.   If you see an error message that the server name or address cannot be resolved, but you can access the web application through Internet Explorer while not recording a session, you might have to configure your proxy settings. In the Actions pane, under Web Application, you can click Configure Settings for any website to select authentication settings.  5. Optionally, add requests or edit captured requests by using the Insert request or the Properties options in the Actions pane.  For more information, see [How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e). |

See Also

[Troubleshooting Web Capture](#z4355df5cdb7e480594dd3e5b5281389b)

Troubleshooting Web Capture

When you select an option to capture a browser session in Operations Manager, the Web Recorder should start, and requests should be recorded as you select them in the browser. If Web Recorder does not work correctly, check the following sections to determine a potential cause and to resolve it.

Enabling Third-Party Browser Extensions

Internet Explorer must be configured to allow third-party extensions on the computer that you are using to perform the capture.

To enable third-party browser extensions

|  |
| --- |
| 1. In Internet Explorer, click Tools, and then click Internet Options.  2. Click the Advanced tab.  3. Under Settings, under Browsing, select Enable third-party browser extensions. |

Run As Administrator

If you are using Windows 7 or Windows Server 2008 R2, or if you are using the Windows Vista or Windows Server 2008 operating system with User Account Control (UAC) enabled, Internet Explorer must be running as an administrator, or the requests will not be added to the Web Recorder. Because the Operations console is starting Internet Explorer, you have to start the Operations console as an administrator.

To run the Operations console as administrator

|  |
| --- |
| 1. Click Start, then All Programs, then System Center Operations Manager 2012.  2. Right-click the Operations console icon.  3. Click Run as administrator. |

X64 Version of Internet Explorer

If you are running the Operations console on a 64-bit operating environment, the Web Recorder might not be displayed when Internet Explorer starts because the x64 version of the add-on was registered and is not used in the x86 version of Internet Explorer. You have to use the x64 version of Internet Explorer for the capture process.

To perform the capture with the x64 version of Internet Explorer

|  |
| --- |
| 1. After you start Internet Explorer from the Operations console, close Internet Explorer.  2. Click the Start button, and on the Start menu, point to All Programs, and then click Internet Explorer (64-bit). |

See Also

[How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e)

How to Edit Settings or Requests in a Web Application

You can use the Web Application Editor to manually create or edit a request in a Web Application Transaction Monitoring template. For editing a particular request, there is no difference whether the request was created manually or by capturing a browser session. For detailed information about the properties that you can set for the request, see [Web Application Request Properties](#zf867812bcdde4d6e8e057cfec9f88cb8).

To open the Web Application Editor

|  |
| --- |
| 1. Start the Operations console with an account that has Author credentials in the management group.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then click Web Application Transaction Monitoring.  4. Select the template that you want to edit, and then click Edit Web Application settings.  Use one of the following procedures to edit the settings that you want. |

To edit the settings for a Web Application

|  |
| --- |
| 1. With the Web Application Editor open, set the properties of the application and then click Apply. For detailed information about the different options, see [Web Application Properties](#z1eb0d2cebe9a4bcfbe18a2ae9321adbb). |

To edit an existing request in a Web Application

|  |
| --- |
| 1. With the Web Application Editor open, select the request that you want to edit.  2. In the Actions pane, click Properties.  3. Set the properties of the request and then click OK. For detailed information about the different options, see [Web Application Request Properties](#zf867812bcdde4d6e8e057cfec9f88cb8).  4. Click Apply to save the web application settings. |

To add a request to a Web Application

|  |
| --- |
| 1. With the Web Application Editor open, select the location in the browser session where you want to insert the request.  2. In the Actions pane, click Insert Request.  3. Type the URL of the request in the Request URL box.  4. Optionally, set other properties of the request, and then click OK. For detailed information about the different options, see [Web Application Request Properties](#zf867812bcdde4d6e8e057cfec9f88cb8). |

See Also

[Web Application Properties](#z1eb0d2cebe9a4bcfbe18a2ae9321adbb)

[Web Application Request Properties](#zf867812bcdde4d6e8e057cfec9f88cb8)

[How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e)

Web Application Properties

The following sections describe the settings available for a Web Application Transaction Monitoring template in Operations Manager. You can set the properties of these requests by using the procedure in [How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e). Each section in this topic represents a tab in the Web Application Properties dialog box.

General Tab

Use the General tab to specify the general details of the application. The various options are explained in the following table.

|  |  |
| --- | --- |
| Item | Description |
| Web Application Name | The name of the application that appears in the Operations console. |
| Description | Optional description of the application that appears in the Details pane in the Operations console. |
| Retry Count | Number of times to retry connecting to a site if the first attempt fails. |
| Management Pack | The management pack in which the Web Application Transaction Monitoring template is stored. This cannot be changed. |
| Authentication Method | Specifies the authentication method to use for the website. If no authentication is required, select None. |
| User Account | The Run As account to use for authenticating on the site. Only existing accounts that match the selected authentication method are listed. For more information about Run As accounts, see Managing Run As Accounts and Profiles. |
| Use a proxy server to connect | Select this option if the watcher nodes must connect to the website through a proxy server. |
| Address | The address of the proxy server if one is required. |
| Port | The port for the proxy server if one is required. |
| Authentication Method | Specifies the authentication method to use for the proxy server. If no authentication is required, select None. |
| User Account | The Run As account to use for authenticating on the proxy server. Only existing accounts that match the selected authentication method are listed. For more information about Run As accounts, see Managing Run As Accounts and Profiles. |

Watcher Node Tab

Use the Watcher Node tab to specify the watcher nodes that you want to use for this web application and the frequency that you want to run the web application. For more information about watcher nodes, see [Watcher Nodes](#z63abfb6894494922940c0f4b80f8c4b1).

Performance Criteria Tab

Use the Performance Criteria tab to enable the Transaction response time monitor for the application that monitors for the transaction time of all of the requests in the browser session. The various options are explained in the following table.

|  |  |
| --- | --- |
| Item | Description |
| Error Transaction Response Time | Select this option and provide a criteria and number of seconds if you want to monitor for a critical state. If the time to process the complete set of requests matches this criteria, the monitor is set to a critical state. |
| Warning Transaction Response Time | Select this option and provide a criteria and number of seconds if you want to monitor for a warning state. If the time to process the complete set of requests matches this criteria, and the error criteria is not also true, the monitor is set to a warning state. |

Performance Counter Tab

Use the Performance Counter tab to enable collection of performance counters for the web application. The various options are explained in the following table.

|  |  |
| --- | --- |
| Item | Description |
| Transaction Response Time | If this option is selected, then the collective time to process all the requests in the browser session is collected. |
| Request Performance Counters | Select the performance counters that are collected for the application. Any selected counters are collected as an aggregate for all requests in the browser session. Each will also be added to the list of counters for every request in the browser session. |
| Reduction Factor | Select to reduce the number of counters selected for the web application and the requests. It specifies how many query intervals must be completed before each collection. If the value is 1, the counters are collected every time the browser session is run. If it is 2, the counters are only collected every second time the browser session is run, and so on. |

See Also

[How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e)

[Web Application Request Properties](#zf867812bcdde4d6e8e057cfec9f88cb8)

Web Application Request Properties

The following sections describe the settings available for each request in a Web Application Transaction Monitor template in Operations Manager. You can set the properties of these requests by using the procedure in [How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e). Each section in the following sections represents a tab in the Request Properties dialog box.

General Tab

Use the General tab to specify the general details of the request. The different options are explained in the following table.

|  |  |
| --- | --- |
| Item | Description |
| Request URL | The URL that you are requesting. You can specify whether the protocol will be http or https. |
| HTTP Method | The method to use for the request. Most requests use a GET method. The POST method is typically used when selecting an option to submit information to a website, such as clicking a button to submit a name and password. |
| HTTP Version | The version of HTTP that the request specifies to the receiving website. |
| Request Body | Only enabled when the HTTP method is POST. This is the body of the request that the post submits. |
| Insert Parameter | There is an Insert parameter button for both the Request URL and the Request Body. Use these options to replace part of the text with a variable that is populated from a previous request. For more information, see [How to Replace Parameters in a URL Request](#zbdc68ab7419e4696a75627f5862b02a3). |

HTTP Headers Tab

The HTTP Headers tab is used to define the different fields that will be included in the header of the request. If the request is from a recorded session, it includes the headers that your browser used. If you manually created the request, it includes a default set of headers and values. You can use the Edit button to modify an existing header field or the Add button to add a new field. The Insert parameter options are used to replace part of the text with a variable that is populated from a previous request. For more information, see [How to Replace Parameters in a URL Request](#zbdc68ab7419e4696a75627f5862b02a3).

Performance Counter Tab

The Performance Counter tab lets you select the performance counters that will be collected for the request. Any selected counters are added to the list of counters specified in the Web Application settings which enable the counter for an aggregate of all requests in a browser session. The value for any selected counter is collected every time that the request is made.

Monitoring

Use the Monitoring tab to control certain monitoring settings for the request and to specify the details of the request that will be collected when one of the monitors enters a warning or critical state. You can view this collected information in the State Change Events tab of the Health Explorer for the monitor. The different options are described in the following table.

|  |  |
| --- | --- |
| Item | Description |
| Monitor SSL health on secure sites | If the request is using https, monitors that measure the health of the related Secure Sockets Layer (SSL) certificates. |
| Enable health evaluation and performance collection for resources | If selected, a monitor is enabled that displays the status of the resources for the page. Instead of measuring every resource, the total of all resources is evaluated. If this option is not selected, the resource monitor does not function for the request. |
| Enable health evaluation and performance collection for Internal links | Enables the collection of the status of each internal link and includes internal links in the evaluation of the Links Status Code monitor for the request. An internal link is a link that refers to a location on the same page. |
| Enable health evaluation and performance collection for External Links | Enables the collection of the status of each external link and includes external links in the evaluation of the Links Status Code monitor for the request. An external link is a link that refers to a location outside the current page. |
| Link traversal | Specifies the number of levels of external links to collect. If the value is 0, only the links on the page itself are evaluated. If the value is 1, the links on each target page are evaluated. If the value is 2, the links on those target pages are evaluated, and so on. |
| Process response body | Specifies whether to evaluate the response body. You must select this value if you want to use content matching or parameter extraction for the request to work. You can clear this option if you only want to perform simple tests for the page such as monitoring status code and response time. |
| Response body collection | Specifies whether to collect the body of the request response. Select one of the following options:  **** Always collect if you want to collect the response body any time any monitor for the request enters a warning or error state.  **** Do not collect if you never want to collect the response body.  **** Collect on content match criteria if you want to collect the response body only when the Content Match monitor enters a warning or critical state. |
| Collect headers | If selected, the header of the request is collected. |
| Collect link headers | If selected, the header of each link is collected. |
| Collect resource headers | If selected, the header of each resource is collected. |

Custom Error

The Custom Error tab lets you specify error criteria for the request by using information that is not available in the Request Details pane of the Web Application Editor. You can either provide simple criteria by using a single metric, or you can use multiple metrics to specify complex logic. Use the Insert button to add a criterion or a group specifying AND or OR logic. If the criteria that you specify resolve to true when the request is run, the monitor indicates an error for the web application.

Custom Warning

The Custom Warning tab lets you specify error criteria for the request by using information that is not available in the Request Details pane of the Web Application Editor. You can either provide simple criteria by using a single metric, or you can use multiple metrics to specify complex logic. Use the Insert button to add a criterion or a group specifying AND or OR logic. If the criteria that you specify resolve to true when the request is run, the monitor indicates a warning for the web application.

Extraction Rules

The Extraction Rules tab lets you extract a string of text from the body of the response of the request to use in one or more subsequent requests. For more information, see [How to Replace Parameters in a URL Request](#zbdc68ab7419e4696a75627f5862b02a3).

See Also

[How to Edit Settings or Requests in a Web Application](#zd98d735bf2964cfca92138595e60d37e)

[Web Application Properties](#z1eb0d2cebe9a4bcfbe18a2ae9321adbb)

How to Replace Parameters in a URL Request

When you capture a web application by using the Web Application Editor, it can include unique information in one or more requests that changes each time you connect to the application. This information is typically included in the response to a request and then used by one or more subsequent requests.

For example, an application might create a unique session ID when a user logs on. This session ID must be included in each request after the logon process. Without the correct session ID, each of these requests fails. Because you do not know what this value is until the first request is run, it cannot be explicitly included in the configuration of the request. If you create the web application by recording a browser session, the session ID is collected in the URL of each request. However, when the application is run, the requests fail because the session ID will have a value that is different from the recorded session ID.

To configure such an application, you can extract a context parameter from the body of the response of one request and use the value of that parameter in one or more subsequent requests. You then replace the explicit value in subsequent requests with a variable that represents the parameter. Each time the synthetic transaction is run, the parameter is populated in the request where it is defined. When the variable is used in the subsequent requests, it is replaced with the collected value before the request is sent to the application.

A single application can use any number of context parameters. Any number of requests can use a single parameter but must be run after the request where the parameter is defined.

Session ID Example

Consider the example where an application creates a session ID when a user logs on. This session ID is required in each request after the logon page. To implement this scenario, you have to capture the session ID when it is first generated, and then use that value in each subsequent request.

You start by using the process described in [How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e) to capture the logon and subsequent actions. The recorded session for logging on to the application and performing some actions might look similar to the following example.

http://www.myapp.com/home.aspx

http://www.myapp.com/search.aspx?query=testing&sessionid=32793279321721

http://www.myapp.com/results.aspx?sessionid=32793279321721

http://www.myapp.com/submit.aspx?sessionid=32793279321721

In this request sequence, the session ID is created by the first request and used in the second, third, and fourth requests. When you run this monitor, it fails because the first request generates a new session ID that could not match the session ID that was used when the session was captured.

To configure this request sequence with parameter replacement, you have to create an extraction rule on the first request to create a context parameter for the session ID. The extraction rule inspects the body of the request to locate the value for the sessionid variable. You would then modify the subsequent requests to use this parameter instead of the value for the session ID.

The modified requests look similar to the following example.

http://www.myapp.com/home.aspx

http://www.myapp.com/search.aspx?query=testing&sessionid=$ParametersContext/sessionID$

http://www.myapp.com/results.aspx?sessionid=$ParametersContext/sessionID$

http://www.myapp.com/submit.aspx?sessionid=$ParametersContext/sessionID$

Creating an Extraction Rule

Context parameters are collected by an extraction rule, and each extraction rule collects a single context parameter. You create an extraction rule in the Properties dialog box of the request that initially generates the required data. To identify the value to extract, you must view the body of the response returned from the particular request. You can either view the source of the page returned in the browser or use a tool that lets you inspect the details of HTTP responses. You cannot view the text by using the Web Application Editor.

When you have identified the request that contains the information you have to extract, you view the Extraction Rules tab in the properties of that request and create one or more extraction rules. The details of each extraction rule are shown in the following table.

|  |  |
| --- | --- |
| Option | Description |
| Context parameter name | Enter the name to give the context parameter. |
| Starts with | Enter the text in the body of the response that identifies the start of the parameter value. You should specify enough characters to ensure that the string is unique. The value for the parameter starts immediately after the last specified character. |
| Ends with | Enter the text in the body of the response that identifies the end of the parameter value. The value for the parameter ends immediately before the first specified character. |
| Index | If the text in the Starts with box occurs more than one time, this value indicates which value to use. If the text only appears one time, or the first occurrence of it shows the text to extract, the value should be 0. If the second value should be extracted, the value should be 1, and so on. |
| Ignore case during search for matching text | Specifies whether to ignore the case of the characters being searched by the Starts with and Ends with boxes. |
| Perform URI encoding of extracted strings | Specifies whether to encode the extracted string after it is collected. |

Inserting a Parameter into a Request

You use a parameter in a request by replacing the explicit value with a variable representing the parameter. The format of the variable is $ParametersContext/<ContextParameterName>$. When the request is run, the variable is replaced with the data extracted by the parameter.

You can insert the variable into the request by using one of the two following methods:

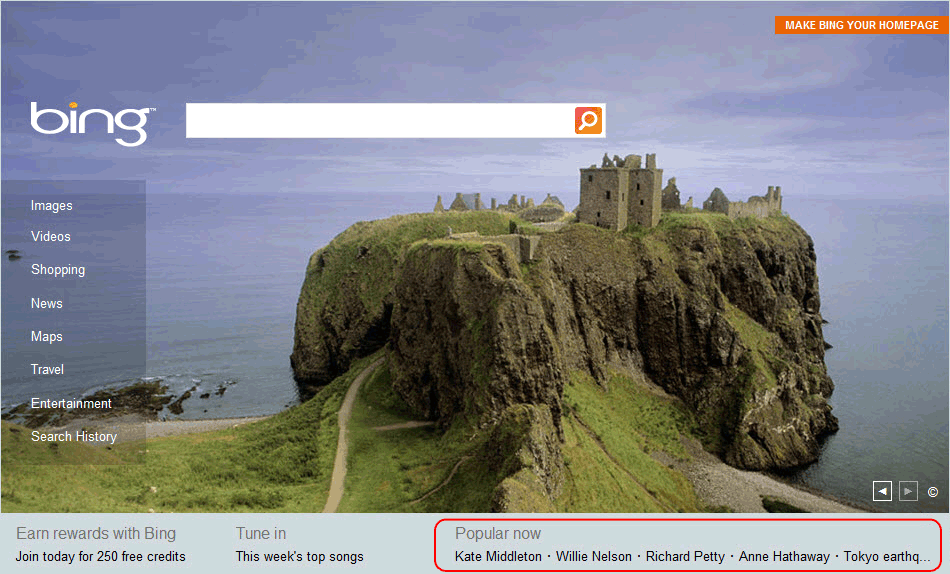
 In the Request Properties dialog box, click the General tab, and then click Request URL to modify the request URL for the request.

 In the Request Properties dialog box, click the General tab, and then click the Insert parameter button. Use the Insert Parameter dialog box for the request. This is accessed from the Insert parameter button on the General tab in the Request Properties dialog box for the request.

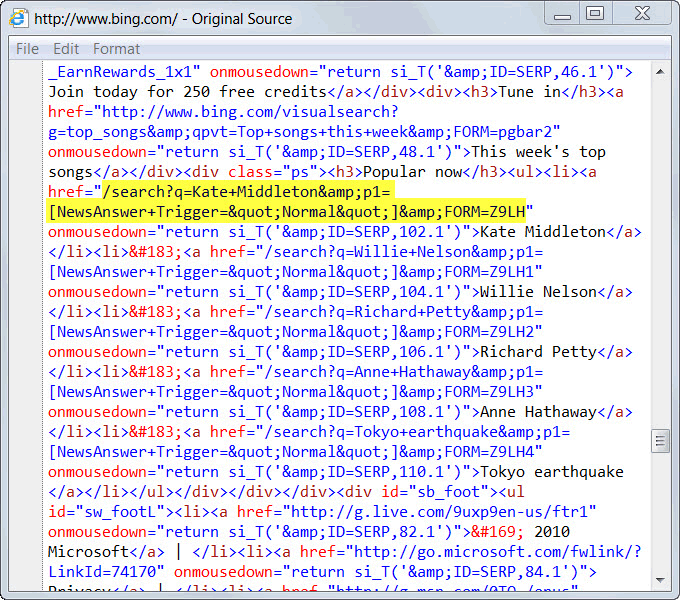
Sample Web Application Using Parameter Extraction

The following procedure provides an example of using parameter extraction in a web application. This example performs a query for the first entry in the Popular Now section of the Bing home page. Because this value changes regularly, you have to first connect to the main page and collect the search term from the body of the response. You then use this term to build the request to perform the actual search.

The main Bing page is shown below with the Popular Now section highlighted.



To determine where in the response body the search term appears, you can view the source of the page. A portion of the source is shown below with the HTML code of the Popular Now section. In this HTML code, you only need the search string which is highlighted in the following illustration. The request is formed from http://www.bing.com followed by this string.



You could just pull out the term itself, but it is more straightforward to include the entire string in the parameter. This string is preceded by the characters <h3>Popular now</h3><ul><li><a href=" and ends with the next occurrence of ". Those are the values that you will use when you define the parameter extraction.

To record a sample web application

|  |
| --- |
| 1. Use the procedure in [How to Capture Web Application Recording](#zcb965b50f31946da96b6ca224c22f61e) to record a web application.  2. While recording, connect to http://www.bing.com.  3. Optionally, use the option on your browser to view the source of the Bing home page and locate the Popular Now section of the HTML code.  4. Click the first search term under Popular Now.  5. Save the recording to the web application.  6. Remove the last request because this is not required. To remove the last request, select the request, and then click Delete in the Actions pane. The resulting requests should look similar to the following URLs: |

To create an extraction rule

|  |
| --- |
| 1. Select the first request, and then click Properties in the Actions pane.  2. Select the Extraction Rules tab.  3. Click Add. The Add Extraction Rule dialog box opens.  4. In the Add Extraction Rule dialog box, in the Context parameter name box, type SearchString.  5. In the Starts with box, type <h3>Popular now</h3><ul><li><a href=".  6. In the Ends with box, type ". The extraction rule should look similar to the following illustration.    7. Click OK to save and close the extraction rule.  8. Click OK to save and close the request. |

To insert a parameter into a request

|  |
| --- |
| 1. Select the second request, and then click Properties in the Actions pane.  2. On the General tab, click Insert parameter.  3. In the String box, delete all text after www.bing.com/.  4. With the cursor positioned at the end of the URL, just after www.bing.com, select SearchString in the Parameters box, and then click Insert. This inserts the variable $ParametersContext/SearchString$. The final request looks similar to the following illustration.    5. Click OK to close the dialog box.  6. Click OK to save and close the request. The modified request sequence should look similar to the following illustration.    7. Click Apply to apply the changes, and then close the Web Application Editor. |

Windows Service Template

The Windows Service template lets you find and monitor instances of a particular service installed on a Windows-based computer. The template locates computers that are running the service and then applies monitors and rules to test its availability and collect performance data. The only information that you have to provide is the name of the service and the types of monitoring that you want to perform.

Scenarios

Use the Windows Service template for any application that uses a service because typically the basic health of the service is critical to the health of the application. You can simply provide the name of the service and have it discovered and monitored on any computer where the application is installed.

Monitoring Performed by Windows Service Template

Depending on your selections in the Windows Service Template wizard, the monitoring performed by the created monitors and rules can include any of the following settings.

|  |  |  |
| --- | --- | --- |
| Type | Description | Enabled? |
| Monitors | Running state of the service | Enabled. |
| CPU utilization of the service | Enabled if CPU Usage monitoring is selected in the wizard. |
| Memory usage of the service | Enabled if Memory Usage monitoring is selected in the wizard. |
| Collection Rules | Collection of events indicating a change in service’s running states. | Enabled. |
| Collection of CPU utilization for the service | Enabled if CPU Usage monitoring is selected in the wizard. |
| Collection of memory usage for the service | Enabled if Memory Usage monitoring is selected in the wizard. |
| Collection of Handle Count for the service | Disabled. Can be enabled with an override. |
| Collection of Thread Count for the service | Disabled. Can be enabled with an override. |
| Collection of Working Set for the service | Disabled. Can be enabled with an override. |

Wizard Options

When you run the Windows Service template, you have to provide values for options in the following tables. Each table represents a single page in the wizard.

General Options

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the service. This name is displayed in the Operations console for the wizard. |
| Description | Optional description of the service. |
| Management Pack | Management pack to store the class and monitors that the template creates. If you create any additional monitors or rules that use the service as a target class, they have to be stored in the same management pack.  For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

Service Details

The following options are available on the Service Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Service name | The name of the service. This name is searched on the agent-managed computer to determine whether it is installed. |
| Targeted group | The service is only discovered on computers that are included in the specified group. |
| Monitor only automatic service | If selected, only those services that are set to start automatically when Windows starts are monitored. Any services with their startup value set to manual or anything other than Automatic are not monitored. |

Performance Data

The following options are available on the Performance Data page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Generate an alert if CPU usage exceeds the specified threshold | Specifies if CPU usage should be monitored. A monitor is created to set an error state on the object and generate an alert when the specified threshold is exceeded. A rule is created to collect CPU usage for analysis and reporting. |
| CPU Usage (percentage) | If CPU usage is monitored, this option sets the threshold. If the percentage of total CPU usage exceeds the threshold, the object is set to an error state and an alert is generated. |
| Generate an alert if memory usage exceeds the specified threshold | Specifies whether memory usage should be monitored. A monitor is created to set an error state on the object and generate an alert when the specified threshold is exceeded. A rule is created to collect CPU usage for analysis and reporting. |
| Memory Usage (MB) | If memory usage is monitored, this option sets the threshold. If the percentage of total CPU usage exceeds the threshold, the object is set to an error state and an alert is generated. |
| Number of samples | If CPU usage or memory is monitored, this option specifies the number of consecutive performance samples that must be exceeded before the object is set to an error state and an alert is generated.  Specifying a number greater than 1 for this option limits the noise from monitoring by ensuring that an alert is not generated when the service only briefly exceeds the threshold. The larger the value that you set, the longer the period of time before you receive an alert. A typical value is 2 or 3. |
| Sample Interval | If CPU usage or memory is monitored, this option specifies the length of time between performance samples.  A smaller value for this option reduces the time for detecting a problem but increases overhead on the agent and the amount of data collected for reporting. A typical value is between 5 and 15 minutes. |

Additional Monitoring

In addition to performing the specified monitoring, the Windows Service template creates a class that you can use for additional monitors and workflows. Any monitor or rule that is using this class runs on any agent where the service is installed. If it creates Windows events that indicate an error, for example, you could create a monitor or rule that detects the particular event and uses the service’ class as a target.

Creating and Modifying Windows Service Templates

To create a Windows Service template

|  |
| --- |
| 1. Determine the target group for the monitor by using the following logic:   If you want to discover the service on all Windows-based computers in the management group, you do not have to create a group. You can use the existing group All Windows Computers.   If you only want the service to be discovered on a certain group of computers, either ensure that an appropriate group exists or create a new group by using the procedure in How to Create Groups in Operations Manager 2012.   If the service you are monitoring is in a cluster, create a group with objects of the class Virtual Server representing the nodes of the cluster that contains the service.  2. Start the Add Monitoring wizard.  3. On the Select Monitoring Type page, select Windows Service, and then click Next.  4. On the General Properties page, in the Name and Description boxes, type a name and description for this new monitor.  5. Select a management pack in which to save the monitor, or click New to create a new management pack. For more information, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  6. Click Next.  7. In the Service Name box, type the name of the specific service that you want to monitor, or click the ellipse (…) button to browse for the service. You can select any computer that has the service installed.  8. Under Targeted Group, specify the group from step 1 of this procedure.  9. Clear the Monitor only automatic services option if you want the monitor to apply to services that are not configured to start automatically. If the service that you are monitoring is in a cluster, clear this option.  10. Click Next.  11. Select the performance counters and thresholds that you want to monitor. For more detailed information, see the [Wizard Options](#z62) section.  12. If you have selected performance counters, specify the monitoring interval.  13. Click Next.  14. Review the summary of the monitor, and then click Create. |

To modify an existing Windows Service template

|  |
| --- |
| 1. Open the Operations console with a user account that has Author credentials.  2. Open the Authoring workspace.  3. In the Authoring navigation pane, expand Management Pack Templates, and then select Windows Service.  4. In the Windows Service pane, locate the monitor to change.  5. Right-click the monitor, and then select Properties.  6. Enter the changes that you want, and then click OK. |

Viewing Windows Service Monitors and Collected Data

To view all Windows Service monitors

|  |
| --- |
| 1. Open the Operations console.  2. Open the Monitoring workspace.  3. In the Monitoring navigation pane, select Windows Service and Process Monitoring, and then click Windows Service State. |

To view the state of each monitor

|  |
| --- |
| 1. In the Windows Service State pane, right-click an object. Select Open, and then click Health Explorer.  2. Expand the Availability and Performance nodes to view the individual monitors. |

To view the performance collected for a service

|  |
| --- |
| 1. In the Windows Service State pane, right-click an object. Select Open, and then click Performance.  2. In the Legend pane, select the counters that you want to view.  3. Use options in the Actions pane to modify the Performance view. |

Monitors and Rules

Monitors and rules are the primary elements for measuring health and detecting errors in Operations Manager and provide similar yet distinct functionality. Monitors set the state of an object while rules create alerts and collect data for analysis and reporting. Each monitor and rule is primarily defined by the source of the data that is used to perform its required functionality and the logic used to evaluate this data.

Although they provide different functionality, monitors and rules both use a common set of sources that provide the data to evaluate. For example, a monitor may use a performance counter to set the state of a particular object. A rule may access the same performance counter in order to store its value for analysis and reporting.

Monitors

A monitor measures the health of some aspect of a managed object. There are three kinds of monitors as shown in the following table:

Health State

Monitors each have either two or three health states. A monitor will be in one and only one of its potential states at any given time. When a monitor loaded by the agent, it is initialized to a healthy state. The state will change only if the specified conditions for another state are detected.

The overall health of a particular object is determined from the health of each of its monitors. This will be a combination of monitors targeted directly at the object, monitors target at objects rolling up to the object through a dependency monitor, dependency monitors targeted at those objects, and so on. This hierarchy is illustrated in the Health Explorer of the Operations console. The policy for how health is rolled up is part of the configuration of the aggregate and dependency monitors.

When you create a monitor, you must specify a condition for each of its health states. When one of those conditions is met, the monitor changes to that state. Each of the conditions must be unique such that only one can be true at a particular time. When a monitor changes to a Warning or Critical state, then it can optionally generate an alert. When it changes to a Healthy state, then any previously generated alert can optionally be automatically resolved.

Types of Monitors

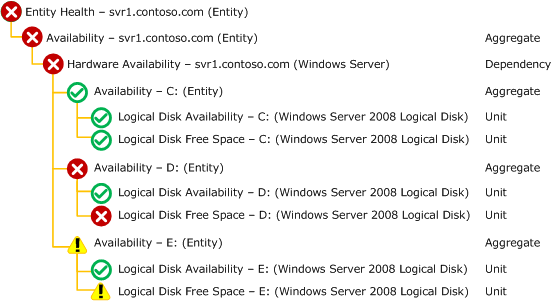
|  |  |
| --- | --- |
| Monitor | Description |
| Unit Monitors | Measures some aspect of the application. This might be checking a performance counter to determine the performance of the application, running a script to perform a synthetic transaction, or watch for an event that indicates an error. Classes will typically have multiple unit monitors targeted at them to test different features of the application and to monitor for different problems. |
| Dependency Monitors | Provides health rollup between different classes. This allows the health of an object to depend on the health of another kind of object that it relies on for successful operation. |
| Aggregate Monitors | Provides a combined health state for similar monitors. Unit and dependency monitors will typically be configured under a particular aggregate monitor. In addition to providing better general organization of the many different monitors targeted at a particular class, aggregate monitors provide a unique health state for different categories of the class. |

Note

When the term monitor is alone, it typically refers to a unit monitor. Aggregate and dependency monitors will typically be referred to with their full name.

The following diagram shows an example of the Health Explorer for the Windows Server class. This shows the use of the different kinds of monitors contributing to an overall health state.

Sample Health Explorer



Rules

Rules do not affect the health state of the target object. They are used for one of three functions as described in the following table:

|  |  |
| --- | --- |
| Monitor | Description |
| Alerting Rules | Alerting rules create an alert when a particular event is detected. This can be any of the different kinds of events shown in [Data Sources](#z81c5ec95966d440ea431398d7b8be96e). Alerting rules do not affect health state. |
| Collection Rules | Collection rules collect events or performance data to the Operations Manager database and data warehouse. They do not create alerts or set health state.  Collection rules can use each of the data sources shown in [Data Sources](#z81c5ec95966d440ea431398d7b8be96e). The data will be converted to either an event or to performance data depending on the particular kind of collection rule you are creating. For some wizards, you will need to provide the specific information from the data source that will be used for different properties of the data being collected. |
| Command Rules | Command rules will run a script or command on a schedule. They do not create alerts or collect any data. |

Should you create a monitor or a rule?

Unit monitors and rules in Operations Manager are similar. They are both workflows that run on an agent, they both can generate an alert when a particular condition is met, and they both use a similar set of data sources to detect these conditions. As a result, it can be difficult to determine if you want to create a monitor or rule for a particular scenario.

Use the following criteria to determine which one to create for different conditions.

Create a monitor if…

 You want to affect the health of an object. In addition to generating an alert, a monitor will affect the health state of its target object. This is displayed in state views and availability reports.

 You want to automatically resolve an alert when the error condition has been cleared. An alert from a rule cannot be automatically cleared since a rule has no way of detecting that the problem has been resolved. A monitor can detect that the problem has been resolved when the condition for its healthy state is met, and the alert can automatically be resolved.

 You are creating an alert based on a performance threshold. There are no rules available to generate an alert from a performance threshold. A monitor should be used for this scenario anyway since you can use the condition where the performance counter is under the defined threshold.

 You have a condition that requires more complex logic than is possible with rules. The Operations console provides a variety of options for setting the health state of a monitor but only simple detection for a rule. If you need more complex logic for a rule but don’t have a method to detect the monitor’s healthy state, then you can create a monitor using Manual or Timer reset. See [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) for more information.

Note

Using the [Viso Management Pack Designer](#z2) you can create custom rules using the same logic available in the Operations console for monitors.

Create a Rule if…

 You want to collect performance counters or events for analysis and reporting. Monitors only collect this information when it initiates a change in health state. If you want to collect the information you need to create a collection rule.

If you want to both collect a performance counter and set a threshold for it to set a health state, then create both a rule and a monitor using the same performance counter.

 You want to generate an alert that is not related to health state of an object.

Monitors and Rules Topics

Monitors and rules are described in the following topics.

 [Data Sources](#z81c5ec95966d440ea431398d7b8be96e)

Describes the concept of a data source and lists the different kinds of data sources available for monitors and rules.

 [Expressions](#z5e4020ce75c54c178719baeeafd546a5)

Describes how to create an expression for different kinds of monitors and rules.

 [Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

Describes how to configure alerts created by monitors and rules.

 [Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

Describes monitors and rules that use different kinds of events and provides details and procedures for creating them using wizards in the Operations console.

 [Performance Monitors and Rules](#z7d91fa3c13794325a87bbef3aa7def9d)

Describes monitors and rules that collect and monitor performance and provides details and procedures for creating them using wizards in the Operations console.

 [Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28)

Provides the details of how to write a monitoring script and how to create monitors and rules using scripts.

 [Dependency Monitors](#zf7e18fe66b1b41e2a39274cc3f220c43)

Describes monitors that allow the health of one kind of object to be dependent on the health of another object.

 [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a)

Describes monitors that consolidate the health of other monitors for a particular kind of object.

Data Sources

Monitors and rules in Operations Manager each start with a data source which defines where it will get the data to evaluate or collect. The first decision to make when you are defining a rule or monitor is the data source that it will use. The most straightforward method to answer this question is to determine where the information is that you want to collect or that indicates the condition you want to detect.

For example, an application may create an event in the Windows Event Log when a particular error occurs. You could create a rule that watches for this particular event and generates an alert when it detects one. If detection of the event is not possible through an event, a log, or a performance counter, then you may need to run a script on a periodic basis to retrieve the required information.

The same set of data sources is available for both monitors and rules as shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Data Source | | Description |
| [Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e) | [Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995) | | Events in the Windows event log matching specified criteria. |
| [Text Logs](#za3d2ce59a94b4dd08aeb5f985c56a964) | | Text log file that has a single line per entry. |
| [WMI Events](#z3474a6f4bca94492a6534459e19b1d2a) | | Events created by Windows Management Instrumentation (WMI). |
| [SNMP Events](#zc6c5bcfa32e6400d8256c083042f18a4) | | Traps sent from an SNMP device. |
| [Syslog Events](#z1a4516f7792b48089232cced8fc44035) | | Events from Unix systems and other devices. |
| [Performance Monitors and Rules](#z7d91fa3c13794325a87bbef3aa7def9d) | [Windows Performance Collection Rules](#z5f5058960954403dbdc819e372212c6c) | | Monitor a threshold or collect a performance value from Windows. |
| [WMI Performance](#zc078fa8e264440499efc0ee1e589d5ab) | Monitor a threshold or collect a performance value from a WMI query. |
| [Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28) | [Script Monitors](#ze411a40fa71e47a99b41f5f1f8bfcfe4) | Monitor a value from a script that runs on a schedule. |
| [Script Collection Rules](#z548dcbc638464d089268c55de8fbdf57) | Collect events or performance data from a script that runs on a schedule. |

Expressions

Wizards for creating monitors and rules will often require you to enter an expression that specifies criteria for the data being collected. The monitor or rule will only apply if the expression is true.

For example, you might have a rule that creates an alert for a particular event. You don’t want an alert for every single event that writes to the event log, so you specify the event number and event source in the expression. The rule will analyze all events that are written to the event log, but it will only generate an alert for those events with the specified source and number.

An expression may be simple with only a single criterion, or it may be a compound expression with multiple criteria and complex logic. Most expressions that you create will have only one or two criteria with very few using complex logic.

The syntax that you use for the expression will be different for different kinds of data sources. For some data source, you will be able to select criteria for a dialog box which keeps you from having to understand the underlying syntax. For other data sources, you will have to know the appropriate syntax and type it in. The following sections provide you with the details of the criteria for each data source.

Criteria Syntax

A single piece of criteria is comprised of a Parameter Name, an Operator, and a Value. Each of these is described in detail in the following sections.

Parameter Name

The parameter name specifies a parameter from the data source for the rule or monitor. The syntax of the parameter name will be different depending on the type of data being collected. The syntax of the parameter name will be different depending on the type of data being collected.

The sections below provide the parameter name syntax for different kinds of data sources.

Windows Events

Windows events provide a prompt in the expression dialog box to select individual properties so you will typically not have to understand the actual syntax. The list of properties with their description is at [Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995).

Event Description is not included in the dropdown list for property name. It can be used by typing in EventDescription. Before using Event Description though, you should verify whether the information that you are using in the description is available in parameters. Event descriptions are often made up of standard text with unique information included through parameters. Parameters are more efficient that the full description since they contain a specific piece of information.

Text Logs

Text Logs do not provide a prompt for the parameter name, so you need to type it in using the appropriate syntax. There are two types of text logs: Generic Text Logs where each line is processed as a single entry and Generic CSV Text Logs which use a delimiter to separate the fields of each entry.

For a Generic Text Log, the entire entry is referred to with a single parameter using the following syntax:

Params/Param[1]

For a Generic CSV Text Log, each field of the entry is referred to with a separate parameter for each field of the entry using the following syntax where # refers to the number of the field starting with 1:

Params/Param[#]

Further details on text log expressions are available at [Text Log Expressions](#z65).

WMI Events

WMI Events do not provide a prompt for the parameter name, so you need to type it in using the appropriate syntax.

The properties available for a WMI event will vary, depending on the kind of event being monitored. The data will be in the form of a property bag that has a collection of properties for one or more WMI class instances. WMI events created by using a query that uses either \_\_InstanceCreationEvent or \_\_InstanceDeletionEvent will have a single collection called TargetInstance with the instance being either created or deleted. WMI events created by using \_\_InstanceModificationEvent will have an additional collection called PreviousInstance.

The syntax for properties from a WMI event is as follows:

Collection[@Name='TargetInstance']/Property[@Name='Caption']

Further details on WMI Events are available at [WMI Events](#z3474a6f4bca94492a6534459e19b1d2a).

Syslog Events

Syslog Events do not provide a prompt for the parameter name, so you need to type it in using the appropriate syntax. The syntax for the properties of a syslog event is simply the name of the property. These properties are listed in [Syslog Events](#z1a4516f7792b48089232cced8fc44035).

SNMP Events

SNMP probes and traps do not provide a prompt for the parameter name, so you need to type it in using the appropriate syntax. The syntax for the properties in the header of an SNMP probe or trap is simply the name of the property.

When a single OID is used:

SnmpVarBinds/SnmpVarBind/ElementName

When you have multiple OIDs and want to refer to each by its numeric order. The first OID is 1, the second is 2, and so on:

SnmpVarBinds/SnmpVarBind[#]/ElementName

When you have multiple OIDs and want to refer to each by the specific OID:

SnmpVarBinds/SnmpVarBind[OID="OID"]/ElementName

Further details on SNMP events are listed in [SNMP Events](#zc6c5bcfa32e6400d8256c083042f18a4).

Scripts

Scripts do not provide a prompt for the parameter name, so you need to type it in using the appropriate syntax. Monitoring scripts output information in the form of a property bag that includes one or more values. The parameter specifies the name of one of the values from the property bag using the following syntax:

Property[@Name="PropertyName"]

Further details on monitoring scripts are available at [Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28).

Operator

The operator specifies the comparison that will be performed between the value from the data property specified in Parameter Name and the value specified in Value. Possible values are shown in the following table.

|  |  |
| --- | --- |
| Operator | Description |
| Equals | The string or number specified in the data is exactly equal to the string or number specified in Value. If this is a string value, the comparison is not case sensitive. |
| Does not equal | The string or number specified in the data is not exactly equal to the string or number specified in Value. If this is a string value, the comparison is not case sensitive. |
| Greater than | The value in the data is greater than the number specified in Value. |
| Greater than or equal to | The value in the data is greater than or equal to the number specified in Value. |
| Less than | The value in the data is less than the number specified in Value. |
| Less than or equal to | The value in the data is less than or equal to the number specified in Value. |
| Contains | The string specified in Value appears somewhere in the data. |
| Does not contain | The string specified in Value does not appear somewhere in the data. |
| Matches wildcard | The string specified in Value matches the string including wildcard. The wildcard character is \* and represents any number of characters. |
| Does not match wildcard | The string specified in Value does not match the string including wildcard. The wildcard character is \* and represents any number of characters. |
| Matches regular expression | The string in the data matches the regular expression specified in Value. |
| Does not match regular expression | The string in the data does not match the regular expression specified in Value. |

Value

The value can be specific text or a number typed into the Value field. For example, a particular event might be defined by its source and number. These are both constant values that can be typed into the Value field.

A value can also come from a property on the target object. Any property on the target object or on any of the object’s parents can be used. You can view a list of the properties and their values for any object by viewing the object in the Discovered Inventory view.

Target properties have different values for different objects. For example, you might use Logical Disk (Server) as a target and require the total size of the disk in the criteria. Logical disks have a property called Size (Mbytes) that stores the total size of the disk. The value of this property will be different for different disks in the management group. When you use a target variable for the value, it will be evaluated separately for each object.

You can select a target property by clicking the ellipse button on the right of the criteria line. This will display a list of all available properties for the object that you selected for the target and that objects parents. If you select one of these properties, the appropriate target variable will be added to the criteria.

Examples

Windows Events

The following expression identifies a Windows event with a source of Contoso and an event number of 100.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| AND group (all of these are true) | | |
| Event ID | Equals | 100 |
| Event Source | Equals | Contoso |

The following expression identifies a Windows event with a source of Contoso, an event number of 100, and the word “Error” in parameter 1.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| AND group (all of these are true) | | |
| Event ID | Equals | 100 |
| Event Source | Equals | Contoso |
| Parameter 1 | Equals | Error |

The following expression identifies a Windows event with a source of Contoso, an event number of 100, and the word “Error” anywhere in the description.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| AND group (all of these are true) | | |
| Event ID | Equals | 100 |
| Event Source | Equals | Contoso |
| EventDescription | Contains | Error |

Text Logs

The following expression identifies an entry in a generic text log that contains the word “Error”.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| Params/Param[1] | Contains | Error |

The following expression identifies an entry in a generic csv text log that contains the word “Error” in the third field.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| Params/Param[3] | Equals | Error |

Scripts

The following expression identifies a numeric value from a script called “PerfValue” that is between 10 and 20.

|  |  |  |
| --- | --- | --- |
| Parameter Name | Operator | Value |
| AND group (all of these are true) | | |
| Property[@Name="PerfValue"] | Greater than | 10 |
| Property[@Name="PerfValue"] | Less than | 20 |

Alerts

Alerts in Operations Manager can be generated from either monitors or rules. While the Operations console does not distinguish between each type, there are distinct differences between the two that you should understand when defining monitors and rules. The sections below provide the alert properties that you will need to define when you configure a monitor to generate an alert or if you create an alerting rule.

Alerts from Monitors

An alert will only be generated from a monitor if each of the following is true:

 The monitor is configured to generate an alert.

 The health state of the monitor changed from a Healthy State to a Warning or Error state, depending on the possible health states of the monitor.

 An open alert does not already exist for the same object created by the same monitor.

Alerts are only generated from a monitor when the health state of the monitor changes from a Healthy state. Even though the criteria for the error condition may occur multiple times, multiple alerts will not be generated once the health state of the monitor is set to Warning or Critical. Only after the health state of the monitor returns to Healthy and the error condition occurs again will a new alert be generated.

For example, consider a Windows Event monitor that is configured to set a Critical state when an event with number 101 is detected and reset the monitor when an event with number 100 is detected. When the first event 101 is created, the monitor is set to a Critical state and an alert is generated. Even though you may close the alert, if an additional event 101 is detected a new alert is not created because the monitor did not change its state. Only after the monitor is reset, either by detecting an event 100 or by you manually resetting it, and an event 101 is detected will an alert be generated.

Alert Name

The name of the alert is a single line of static text and cannot include any variables.

Priority and Severity

The Alert severity defines the alert as either Information, Warning, or Critical. This severity does not have to match the severity of the health state triggering the alert. The severity of the alert is identified by an icon in the Operations console and is used by views and notification subscriptions. The alert priority is inaccessible in the Operations console but is used primarily for notification subscriptions.

Alert Description

The alert description may have several lines of text that includes a combination of static text and variables. The most common kind of variable in the alert description will be $Data variables to include different information from the monitor’s data source in the description of the alert. The properties that are available will depend on the kind of data source being used.

The table below provides syntax and examples of variables in alerts created from monitors.

|  |  |  |
| --- | --- | --- |
| Data Source | Syntax | Examples |
| Windows Event | $Data/Context/<Property Name>$ | $Data/Context/EventDescription$ |
| $Data/Context/Params/Param[#]$ | $Data/Context/Params/Param[2]$ |
| Text Log | $Data/Context/<Property Name>$ | $Data/Context/LogFileName$ |
| $Data/Context/Params/Param[1]$ | $Data/Context/Params/Param[1]$ |
| Delimited Text Log | $Data/Context/<Property Name>$ | $Data/Context/LogFileName$ |
| $Data/Context/Params/Param[#]$ | $Data/Context/Params/Param[2]$ |
| WMI Event | $Data/Context/Collection[@Name='<TargetInstance|PreviousInstance>']/Property[@Name='<PropertyName>']$ | $Data/Context/Collection[@Name=’TargetInstance’]/Property[@Name='Name']$ |
| Windows Performance | $Data/Context/<PropertyName>]$ | $Data/Context/Value$ |
| WMI Performance | $Data/Context/<PropertyName>]$ | $Data/Context/Value$ |
| Monitoring Script | $Data/Context/Property[@Name='<PropertyName>']$ | $Data/Context/Property[@Name='Result'>']$ |

Automatic Alert Resolution

Monitors that create alerts can be configured to automatically resolve the alert when the monitor returns to a healthy state. This means that any unresolved alert for the monitor represents a problem that still exists. There is no configuration this requirement other than confirming the option that automatic resolution be performed.

Note

Automatic alert resolution cannot be performed with rules because rules do not have any means of detecting that the problem has been corrected.

Alerts from Rules

An alert will only be generated from a rule under the following conditions:

 The rule is configured to generate an alert.

 The criteria defined in the rule is true.

 An open alert does not already exist that matches the alert’s suppression configuration.

The table below discusses the ability of each type of rule to generate an alert.

|  |  |
| --- | --- |
| Rule Type | Alert Capabilities |
| Event Rules | Alert rules can be created for each event data source. The criteria that is specified to determine when an alert should be created is the same as the criteria for a state change in the event monitors. |
| Performance Rules | You cannot create an alert rule based on a performance counter. A monitor should be used instead because a success condition is usually detectable from a performance counter and is usually related to some health state of the target class. |
| Scripting Rules | You cannot create an alert rule based on a script. A monitor should be used instead because a script will typically provide a return value for both and error and a healthy state in such a way that a success condition is usually detectable and related to some health state of the target class. |

Alert Name

The name of the alert is a single line of static text and cannot include any variables.

Priority and Severity

The Alert severity defines the alert as either Information, Warning, or Critical. This severity does not have to match the severity of the health state triggering the alert. The severity of the alert is identified by an icon in the Operations console and is used by views and notification subscriptions. The alert priority is inaccessible in the Operations console but is used primarily for notification subscriptions.

Alert Description

The alert description may have several lines of text that includes static text or variables. The most common kind of variable in the alert description will be $Data variables to include different information from the rule’s data source in the description of the alert. The properties that are available will depend on the kind of data source being used. Each section of [Data Sources](#z81c5ec95966d440ea431398d7b8be96e) includes a list of the properties available for different data sources.

The following table provides syntax and examples of variables in alerts created from rules:

|  |  |  |
| --- | --- | --- |
| Data Source | Syntax | Examples |
| Windows Event | $Data/<Property Name>$ | $Data/EventDescription$ |
| $Data/Params/Param[#]$ | $Data/Params/Param[2]$ |
| Text Log | $Data/EventData/DataItem/<PropertyName>$ | $Data/EventData/DataItem/LogFileName$ |
| $Data/EventData/DataItem/Params/Param[1]$ | $Data/EventData/DataItem/Params/Param[1]$ |
| Delimited Text Log | $Data/EventData/DataItem/<PropertyName>$ | $Data/EventData/DataItem/LogFileName$ |
| $Data/EventData/DataItem/Params/Param[#]$ | $Data/EventData/DataItem/Params/Param[2]$ |
| WMI Event | $Data/EventData/DataItem/Collection[@Name='<TargetInstance | PreviousInstance>']/Property[@Name='<PropertyName>']$ | $Data/EventData/DataItem/Collection[@Name='TargetInstance']/Property[@Name='Name']$ |
| Syslog Event | $Data/EventData/DataItem/<PropertyName>$ | $Data/EventData/DataItem/Facility$ |

Alert Suppression

Alert suppression refers to logic that is defined on alert rules to suppress the creation of an alert when a corresponding alert is still open. This prevents alert storms where multiple alerts are created for the same issue. Because the issue has already been identified with an open alert, creation of additional alert creates unnecessary noise with minimal value. When the condition for an alerting rule is met but an existing alert is already open, instead of creating an additional alert suppression will increase the repeat count of the existing alert.

In order to define suppression on an alerting rule, the fields must be specified that identify a matching alert. Before an alerting rule creates a new alert, it will check whether an open alert exists with values for the fields that are defined for suppression that match the values in the same fields of the new alert. If an alert with matching values for each of these fields is open, then a new alert is not created.

The minimum number of fields that uniquely identify the alert should be specified for alert suppression. This will typically be the computer name in addition to the fields used for the criteria of the rule. For example, suppression on event rules can frequently be achieved by using the following fields:

 Logging Computer

 Event Source

 Event Number

If the rule is targeted at a class that has multiple instances on an agent, however, then a parameter might be required to uniquely identify the event in the criteria of the rule. If this is the case, then the same parameter should be specified in the alert suppression.

Note

Alert suppression is not available for monitors because it is not required. Monitors only generate alerts when their state changes from healthy to warning or critical. Even if the detected issue occurs again when the monitor is already in the negative state, then no alert is generated because the monitor state does not change. A new alert is only generated if the monitor returns to a healthy state before the problem occurs.

Event Monitors and Rules

Event monitors and rules rely on the application they are monitoring to create an event of some kind in response to a problem or other interesting occurrence. The monitor or rule continuously watches the data source for an event matching specific criteria and immediately takes an appropriate response. The basic logic and configuration of event rules and monitors are similar except for the initial configuration of the data source that they are retrieving the event from.

Types of Event Monitors and Rules

The table below lists the kinds of events that can be used for monitors and rules in an Operations Manager management pack. Each is discussed in more detail in their own topic.

 [Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

Events in a Windows event log.

 [Text Logs](#za3d2ce59a94b4dd08aeb5f985c56a964)

Text log file that has a single line per entry. The log can be a simple text log where each line is considered a single entry or a delimited text log where a single character is used to separate different fields of data.

 [WMI Events](#z3474a6f4bca94492a6534459e19b1d2a)

Events created by Windows Management Instrumentation (WMI).

 [Syslog Events](#z1a4516f7792b48089232cced8fc44035)

Events from Unix systems and other devices that send syslog messages.

 [SNMP Events](#zc6c5bcfa32e6400d8256c083042f18a4)

SNMP traps that are sent to an agent or SNMP probes that are periodic requests for information from a device.

 [UNIX/Linux Shell Command Alerts](#zfcfac291c2154361a21715fb08176c87)

Events that are detected through the execution of an UNIX/Linux command, script, or one-line sequence of multiple commands (using pipeline operators).

Windows Events

Many Windows-based applications post information to events in a Windows event log. This could be a standard log such as Application or a log specific to the application being monitored. These events follow a standard format and frequently contain detailed information about the particular issue. If the application you are monitoring creates a Windows event in response to a particular issue, then this likely be the most effective way to detect the issue in an Operations Manager management pack.

When you create a rule or monitor that uses a Windows event, Operations Manager continuously monitors the log and immediately responds when an event matching the specified criteria is detected. These events are persisted meaning that they are available after they are initially created. Operations Manager will record the last position that it read in the log and continue from that position the next time it reads the log. If the health service on the agent is not running when a particular event is created, Operations Manager will detect it the next time that the agent is started.

Windows Event Wizards

The table below lists the wizards that are available for Windows events.

|  |  |
| --- | --- |
| Management Pack Object | Wizards Available |
| Monitors | Simple Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Repeated Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Missing Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Correlated Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Correlated Missing Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Rules | Alert Generating Windows event rule |
| Event collection Windows event rule |

Windows Event Wizard Options

When you run a Windows event rule or monitor wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule or wizard including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule or monitor. For a rule, the name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. For a monitor, the name appears in the Health Explorer of any target objects. |
| Description | Optional description of the rule or monitor. |
| Management Pack | Management pack file to store the rule or monitor.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category (Rules only) | The category for the rule. For an event collection rule, this should be Event Collection. For an alerting rule, this should be Alert. |
| Parent Monitor (Monitors only) | The aggregate monitor that the monitor will be positioned under in the Health Explorer. For more information, see [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a). |
| Target | The class to use for the target of the rule or monitor. The rule or monitor will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is enabled  Monitor is enabled | Specifies whether the rule or monitor is enabled. |

Event Log Type

The Event Log Type page includes the name of the event log where you expect the event to be created. There will be a single Event Log Type page for a collection or alerting rule and for a monitor using manual or timer reset. For a monitor using Windows Event Reset, you will have to define the log for both the error condition and for the healthy condition. You will typically specify the same log for both conditions, but a different log could be used for each.

You can type in the name of the event log in the Log name box, or you can click the ellipse button and select a log.

Event Expression

In addition to the name of the log to retrieve events from, workflows using a Windows event must specify sufficient criteria to identify the particular events that relate to the issue being identified. Frequently, the Event ID and the Event Source will be sufficient for this purpose. This depends on the kind of information that the application provides in the particular event in addition to the target that is being used for the monitor. If the class being used as the target for the monitor is expected to have multiple instances on a particular agent, then these two properties are probably insufficient for uniqueness. Unless the criteria included a key property for the target class then the criteria would possibly apply to all instances.

There will be a single Event Log Type page for each Event Log Type Page collection or alerting rule and for a monitor using manual or timer reset. For a monitor using Windows Event Reset, you will have to define the log for both the error condition and for the healthy condition. You will typically specify the same log for both conditions, but a different log could be used for each.

The following table lists the properties available from Windows Events. These properties can be accessed for setting criteria in monitors and rules and can be included in alert descriptions.

|  |  |
| --- | --- |
| Expression | Description |
| Event Source | Source of the event. Generally used in the criteria of the monitor or rule. |
| Logname/Channel | Name of the event log such as Application or System. |
| Logging Computer | Name of the computer logging the event. |
| Event ID | Number of the event. |
| Event Category | Category of the event. |
| Event Level | Severity of the event that uses one of the following values.  Success (0)  Error (1)  Warning (2)  Information (4)  Success Audit (8)  Failure Audit (16) |
| User | Name of the user account that was used to create the event. |
| EventDescription | Full event description. |
| Parameter | Collection of event parameters. |

Auto Reset Timer

The Auto Reset Timer page is only available for timer reset monitors. It allows you to set the time that must pass after the alert is created before the alert is automatically resolved.

Configure Health

The Configure Health page is only available for monitors. It allows you to specify the health state that will be set for each of the events. For a manual reset monitor, the Manual Reset condition will be Healthy, and you can specify whether the Event Raised condition will set the monitor to a Warning or a Critical state. For a Timer Reset or a Windows Event Reset, you can specify the health state set by each event. The first event will typically set the monitor to Warning or Critical while the second event or the timer will set the monitor to Healthy.

Configure Alerts

The Configure Alerts page is only available for monitors and alerting rules. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating Windows Event Monitors

How to create a Windows event monitor

Use the following procedure to create an event monitor in Operations Manager with the following details:

 Runs on all agents with a particular service installed.

 Sets the monitor to a critical state when an event in the Application event log with an event source of EventCreateand an event number of 101 is detected.

 Sets the monitor to a healthy state when an event in the Application event log with an event source of EventCreateand an event number of 102 is detected.

Note

EventCreate is used as the event source so that the EventCreate utility can be used to create a test event. This utility is available on any Windows Computer and creates test events with a source of EventCreate. If you have another method of creating test events, then you can use a different source.

To create an event monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace.  4. Select Management Pack Objects.  5. Right-click Monitors, select Create and Monitor, and then select Unit Monitor.  6. On the Monitor Type page, do the following:  a. In the Select the type of monitor to create box, expand Windows Events and then Simple Event Detection.  b. Select Windows Event Reset.  c. In the Management Pack dropdown list, select the management pack for the application.  d. Click Next.  7. On the General page, do the following:  a. In the Name box, type Error event 101 or another name for the monitor. This is the text that will appear in the Health Explorer.  b. Click Select.  c. In the Select Items to Target dialog box, select the name that you used for the Windows Service template in step 2.  d. The Parent monitor box should show Availability. You can select a different parent monitor.  e. Ensure that Availability is selected for the Parent monitor.  f. The Monitor is enabled box should be checked so that the monitor is enabled.  g. Click Next.  8. On the Event Log (Unhealthy Event) page, do the following:  a. In the Log Name box, keep the default value of Application.  b. Click Next.  9. On the Event Expression (Unhealthy Event) page, do the following:  a. For the Event ID value, type 101  b. For the Event Source value, type EventCreate  c. Click Next.  10. On the Event Log (Healthy Event) page, do the following:  a. In the Log Name box, keep the default value of Application.  b. Click Next.  11. On the Event Expression (Healthy Event) page, do the following:  a. For the Event ID value, type 102  b. For the Event Source value, type EventCreate  c. Click Next.  12. On the Configure Health page, do the following:  a. For FirstEventRaised, change the Health State to Critical.  b. For the Event Source value, type EventCreate  c. Click Next.  13. On the Configure Alerts page, do the following:  a. Select Generate alerts for this monitor.  b. Click Create. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Event Monitor Logic](#z9b4a1aa86ed34a34b7f3e6cb18962381)

[Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

Text Logs

A text log is a text file that an application uses to log event information. In order to use a text log data source in a management pack, each entry in the log must be on a single line. If the log file does not fit this requirement, then a [Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28) has to be created to read the log.

Applications that use log files frequently create a new file each day or when one file reaches a certain size. To support this functionality, monitors and rules specify a Directory and a Pattern for the text logs being monitored. Directory is the path of the directory where the text logs will be located. This must be an absolute path without wildcard characters. A $Target variable could also be used if the path to the log files is stored in a property of the target class. Pattern is the name of the log file including wildcard characters as appropriate.

For example, an application might create a log file each day with the date included in the name as in log20100316.txt. A pattern for such a log might be log\*.txt which would apply to any log file following the application’s naming scheme.

A text log can be defined as either a Generic Text Log or Generic CSV Text Log. CSV refers to Comma Separated Values, although you can specify any character as the delimiter. The two kinds of files are also referred as a Simple Text Log and a Delimited Text Log. With a simple text log, each line is considered a single entry. With a delimited text log, a single character is used to separate different fields of data, and each field can be analyzed separately.

Text Log Wizards

The table below lists the wizards that are available for both simple and delimited text files.

|  |  |
| --- | --- |
| Management Pack Object | Wizards Available |
| Monitors | Simple Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Repeated Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Rules | Alert Generating rule |
| Event collection rule |

Text Log Wizard Options

When you run a text log wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule or wizard including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule or monitor. For a rule, the name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. For a monitor, the name appears in the Health Explorer of any target objects. |
| Description | Optional description of the rule or monitor. |
| Management Pack | Management pack to store the rule.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category (Rules only) | The category for the rule. For a collection rule, this should be Event Collection. For an alerting rule, this should be Alert. |
| Parent Monitor (Monitors only) | The aggregate monitor that the monitor will be positioned under in the Health Explorer. For more information, see [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a). |
| Target | The class to use for the target of the rule or monitor. The rule or monitor will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is enabled  Monitor is enabled | Specifies whether the rule or monitor is enabled. |

Application Log Data Source

There will be a single application log data source page for a collection or alerting rule and for a monitor using manual or timer reset. For a monitor using event reset, you will have to define the log for both the error condition and for the healthy condition. You will typically specify the same log for both conditions, but a different log could be used for each.

The following table lists the settings that must be provided for an application log data source:

|  |  |
| --- | --- |
| Property Name | Description |
| Directory | Directory that the log file is located in. This must be a single directory with no wildcards |
| Pattern | Name of the log file. This can include wildcards if the name of the log file will change. Use the ? wildcard to represent a single character. Use the \* wildcard to represent multiple characters. |
| Separator (Delimited Logs only) | The character that is used to separate the |
| UTF8 |  |

Event Expression

There will be a single expression page for a collection or alerting rule and for a monitor using manual or timer reset. For a monitor using event reset, you will have to define an expression for both the error condition and for the healthy condition.

The expression for a text log rule or monitor will include criteria that matches text in the log entry. For a Generic Text Log this includes a search of the whole log entry treated as a single line. For a delimited log file, this will include a search of one or more of the included fields. The contents of a text log are included in the parameters of the event. For a generic text log, this is referenced by the parameter Params/Param[1]. A delimited log uses the same variable by using the index number of the required parameter. The first field would be referenced with Params/Param[1], the second field would be referenced with Params/Param[2], and so on.

The following table lists the common properties available from text log monitors and rules:

|  |  |
| --- | --- |
| Property Name | Description |
| Directory | Directory that the log file is located in. |
| Pattern | Name of the log file that the event was taken from. |
| Param[1] | Complete entry in a generic text log. |
| Param[#] | Specific parameter in a generic CSV text log. # represents the number of the field. |

For more information about expressions, see [Expressions](#z5e4020ce75c54c178719baeeafd546a5).

Auto Reset Timer

The Auto Reset Timer page is only available for timer reset monitors. It allows you to set the time that must pass after the alert is created before the alert is automatically resolved.

Configure Health

The Configure Health page is only available for monitors. It allows you to specify the health state that will be set for each of the events. For a manual reset monitor, the Manual Reset condition will be Healthy, and you can specify whether the Event Raised condition will set the monitor to a Warning or a Critical state. For a Timer Reset or an Event Reset, you can specify the health state set by each event. The first event will typically set the monitor to Warning or Critical while the second event or the timer will set the monitor to Healthy.

Configure Alerts

The Configure Alerts page is only available for monitors and alerting rules. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating Text Log Rules and Monitors

Use the following procedure to create a text log alerting rule in Operations Manager with the following details:

 Runs on all agents with a particular service installed.

 Looks for a comma delimited log file with the naming pattern MyApp\*.log in the c:\logs directory.

 Generates an alert if the word “error” appears in the log message.

 Includes the error message in the alert description.

 The format of each line of the text file is as follows: Date,Time,Message

To create a delimited text log alert rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Rules.  4. Right-click Rules and select Create a new rule.  5. On the Rule Type page, do the following:  a. Expand Alert Generating Rules, expand Event Based, and then click Generic CSV Text Log (Alert).  b. Select the management pack from step 1.  c. Click Next.  6. On the General page, do the following:  a. In the Rule Name box, type MyApplication Delimited Log Error.  b. In the Rule Category box, select Alert.  c. Next to Rule Target click Select and then select the name of the target that you created in step 2.  d. Leave Rule is enabled selected.  e. Click Next.  7. On the Application Log Data Source page, do the following:  a. In the Directory box, type c:\logs.  b. In the Pattern box, type MyApp\*.log.  c. In the Separator box, type a COMMA.  d. Click Next.  8. On the Build Event Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type Params/Param[3].  c. In the Operator box select Contains.  d. In the Value box type error.  e. Click Next.  9. On the Configure Alerts page, do the following:  a. In the Alert name box, type Error found in MyApplication delimited text log..  b. Click the ellipse button to the right side of the Alert description box.  c. Clear the text in the Value box.  d. Select Data, then Params, then Param.  e. Replace the text <<INT>> with 1.  f. Move to the end of the line and press the SPACE bar.  g. Select Data, then Params, then Param.  h. Replace the text <<INT>> with 2.  i. Move to the end of the line and press the ENTER key.  j. Select Data, then Params, then Param.  k. Replace the text <<INT>> with 3.  l. Click OK.  10. Click Finish. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1)

[Repeating Events](#z9dacca1d43be4caaad2a2183c14a0286)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

WMI Events

WMI events are created from WMI queries that detect particular actions in the operating system or in applications that create their own WMI events. These events can be used to detect such actions as a process ending, a file being created, or a registry key being modified. WMI events are not persisted. Therefore, any WMI events that are created when the agent service is not running are lost.

Note

This guide assumes knowledge of how to build a WMI notification query. For a an overview of this topic and sample queries see [Unlocking the Mystery of WMI Events in MOM](http://go.microsoft.com/fwlink/?LinkID=187607).

WMI Event Wizards

The table below lists the wizards that are available for WMI events.

|  |  |
| --- | --- |
| Management Pack Object | Wizards Available |
| Monitors | Simple Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Repeated Event Detection using each of the standard [Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1) methods |
| Rules | Alert Generating WMI event rule |
| Event collection WMI event rule |

WMI Event Wizard Options

When you run a WMI event rule or monitor wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule or monitor including its name, category, target, and the management pack file to store it in.

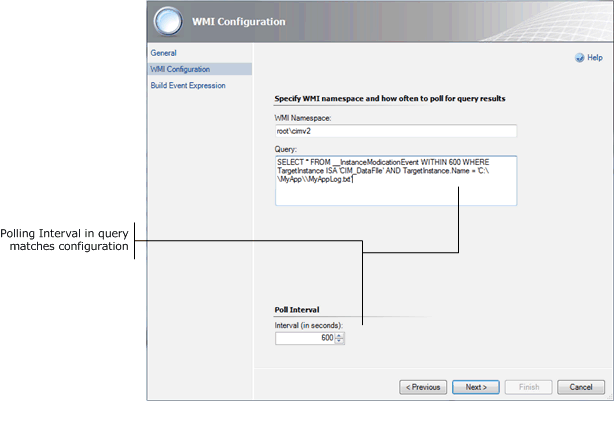
|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule or monitor. For a rule, the name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. For a monitor, the name appears in the Health Explorer of any target objects. |
| Description | Optional description of the rule or monitor. |
| Management Pack | Management pack file to store the rule or monitor.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category (Rules only) | The category for the rule. For an event collection rule, this should be Event Collection. For an alerting rule, this should be Alert. |
| Parent Monitor (Monitors only) | The aggregate monitor that the monitor will be positioned under in the Health Explorer. For more information, see [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a). |
| Target | The class to use for the target of the rule or monitor. The rule or monitor will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is enabled  Monitor is enabled | Specifies whether the rule or monitor is enabled. |

WMI Configuration / WMI Event Provider

The WMI Configuration Page allows you to provide the WMI namespace, query, and poll interval. There will be a single WMI Configuration page for a collection or alerting rule and for a monitor using manual or timer reset. For a monitor using WMI Event Reset, there will be a WMI Event Provider page to define the query for both the error condition and for the healthy condition.

|  |  |
| --- | --- |
| Option | Description |
| WMI Namespace | The namespace containing the class that is used in the WMI query. |
| Query | WMI notification query that looks for the occurrence of a particular WMI event. |
| Poll Interval | Specifies how frequently Operations Manager will poll WMI for the occurrence of the event. This value should be the same as the value used in the WITHIN clause of the notification query. |

WMI matching poll intervals



Build Expression

The Build Expression page allows you to define a filter for the data coming from the WMI query. There will be a single Build Expression page for a WMI event monitor using manual or timer reset. For a monitor using WMI Event Reset, there is an expression for each health state.

Because criteria can be specified in the WHERE clause of the WMI query, an expression is frequently not required in a WMI event monitor. It is only required if the query is expected to return multiple records. WMI event rules rely on the criteria in the query itself and don’t allow an expression. The Operations console wizards though require that criteria be specified in WMI Event monitors. If no criteria is required, then dummy criteria must be specified in the wizard and then removed by viewing the properties of the monitor after it is created.

The properties available for a WMI event will vary, depending on the kind of event being monitored. The properties available will also vary, depending on the properties of the WMI class included in the query. The data will be in the form of a property bag that has a collection of properties for one or more WMI class instances. WMI events created by using a query that uses either \_\_InstanceCreationEvent or \_\_InstanceDeletionEvent will have a single collection called TargetInstance with the instance being either created or deleted. WMI events created by using \_\_InstanceModificationEvent will have an additional collection called PreviousInstance.

The syntax for properties from a WMI event is as follows:

Collection[@Name='TargetInstance']/Property[@Name='Caption']

For example, the following WMI query monitors for the change in a file that is named c:\MyApp\MyAppLog.txt.

SELECT \* FROM \_\_InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'CIM\_DataFIle' AND TargetInstance.Name = 'C:\\MyApp\\MyAppLog.txt'

Assuming that data is added to the file changing the file size and triggering the query, examples of properties from this query are shown in the following table:

|  |  |
| --- | --- |
| Property | Syntax |
| Original file size | Collection[@Name=’PreviousInstance’]/Property[@Name='FileSize'] |
| New file size | Collection[@Name=’TargetInstance’]/Property[@Name='FileSize'] |

Auto Reset Timer

The Auto Reset Timer page is only available for timer reset monitors. It allows you to set the time that must pass after the alert is created before the alert is automatically resolved.

Configure Health

The Configure Health page is only available for monitors. It allows you to specify the health state that will be set for each of the events. For a manual reset monitor, the Manual Reset condition will be Healthy, and you can specify whether the Event Raised condition will set the monitor to a Warning or a Critical state. For a Timer Reset or an WMI Event Reset, you can specify the health state set by each event. The first event will typically set the monitor to Warning or Critical while the second event or the timer will set the monitor to Healthy.

Configure Alerts

The Configure Alerts page is only available for monitors and alerting rules. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating WMI event monitors and rules

The following procedure shows how to create a WMI event monitor in Operations Manager with the following details:

 Runs on all agents with a particular service installed.

 Sets the monitor to a critical state when Notepad is started on the agent computer.

 Sets the monitor to a healthy state when Notepad is ended on the agent computer.

Note

This example is not meant to illustrate a real world scenario since there would be minimal value in monitoring when Notepad is started. It does through represent a common scenario of monitoring two different WMI events in a monitor. Using Notepad provides a sample that is easy to test by starting and stopping Notepad on the agent computer.

To create a WMI event monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace.  4. Right-click Monitors, select Create a Monitor, and then select Unit Monitor.  5. On the Monitor Type page, do the following:  a. Expand WMI Events, then Simple Event Detection, and then WMI Event Reset.  b. Select the management pack from step 1.  c. Click Next.  6. On the General page, do the following:  a. In the Name box, type MyApplication WMI Event Error.  b. Click Select next to the Monitor Target box.  c. Next to Monitor Target click Select and then select the name of the target that you created in step 2.  d. In the Parent Monitor box, select Availability.  e. Leave the Monitor is enabled box checked , select and click Next.  7. On the First WMI Event Provider page, do the following:  a. In the WMI Namespace box, type root\cimv2.  b. In the Query box, type the following WMI query.  Select \* From \_\_InstanceCreationEvent WITHIN 60 Where TargetInstance ISA 'Win32\_Process' and TargetInstance.Name = 'notepad.exe'  c. In the Poll Interval box, type 60.  d. Click Next.  8. On the Build First Expression page, do the following:  Note  In this example, criteria is included in the WMI query, so no expression is required in the monitor. Since the WMI event wizard in the Operations console requires an expression for each event, dummy expressions will be provided to complete the wizard and then deleted once the monitor is created.  a. Click Insert.  b. In the Parameter Name box type Dummy.  c. In the Operator box select Equals.  d. In the Value box type Dummy.  e. Click Next.  9. On the Second WMI Event Provider page, do the following:  a. In the WMI Namespace box, type root\cimv2.  b. In the Query box, paste the following WMI query.  Select \* From \_\_InstanceDeletionEvent WITHIN 60 Where TargetInstance ISA 'Win32\_Process' and TargetInstance.Name = 'notepad.exe'  c. In the Poll Interval box, type 60.  d. Click Next.  10. On the Second Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type Dummy.  c. In the Operator box select Equals.  d. In the Value box type Dummy.  e. Click Next.  11. On the Configure Health page, do the following:  a. Next to FirstEventRaised, change the Health State to Critical.  b. Click Next.  12. On the Configure Alerts page, do the following:  a. Check Generate alerts for this monitor  b. In the Generate an alert when box, select The monitor is in a critical health state.  c. Leave the box selected to automatically resolve the alert.  d. In the Alert name box, type Notepad process detected  e. Click the ellipse button next to Alert description.  f. Clear the contents of the Value box and then type Path of executable: .  g. Click Data, then Collection, then Property.  h. In the variable, replace <<INT>> with "TargetInstance" and <<STRING>> with ExecutablePath. The final text in the Value box should be Path of executable: $Data/Context/Collection["TargetInstance"]/Property[@Name="ExecutablePath"]$  i. Click OK.  13. Click Create.  14. Right-click MyApplication WMI Event Error and select Properties.  15. On the First Expression tab, click Delete.  16. On the Second Expression tab, click Delete.  17. Click OK. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Event Monitor Reset](#zd05005cc56d7434593be09d836da44a1)

[Repeating Events](#z9dacca1d43be4caaad2a2183c14a0286)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

Syslog Events

Syslog events can be used to collect messages from Unix systems and other devices in Operations Manager. Syslog rules can be run on an agent that is the receiver of messages from one or more devices. When the rule is run, the agent will listen for messages on UDP port 514. This is the only port that can be used.

Target

Rules and monitors run on the agent computer of each instance of the target class, and they usually access data on the local computer. SNMP rules and monitors typically work with information from a computer or device different from the one running the monitors or rules. For SNMP traps, the monitor or rule needs to be running on the agent that receives the trap. The device needs to be configured to deliver traps to this agent. For SNMP probes, the monitor or rule needs to be running on any agent that is authorized to access the device with SNMP. The device may need to be configured to allow communication from this agent.

Network devices that are discovered with the Discovery Wizard are managed by a resource pool that you specify during the discovery process. A resource pool contains one or more management servers. You can use the classes for these devices as targets, and the rule or monitor will run on each computer in the resource pool. In this case, the device will need to send SNMP traps to each of the computers in the pool and allow access to each computer in the pool for SNMP probes.

Syslog Event Wizards

The table below lists the wizards that are available for both simple and delimited text files.

|  |  |
| --- | --- |
| Management Pack Object | Wizards Available |
| Monitors | None |
| Rules | Alert Generating rule |
| Event collection rule |

Syslog Event Wizard Options

When you run a Syslog event rule wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule. The name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. |
| Description | Optional description of the rule. |
| Management Pack | Management pack file to store the rule or monitor.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category (Rules only) | The category for the rule. For an event collection rule, this should be Event Collection. For an alerting rule, this should be Alert. |
| Parent Monitor (Monitors only) | The aggregate monitor that the monitor will be positioned under in the Health Explorer. For more information, see [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a). |
| Target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is enabled | Specifies whether the rule is enabled. |

Build Event Expression

The Build Event Expression page allows you to filter for specific events to be collected or to generate an alert. The Syslog data properties are shown in the following table:

|  |  |
| --- | --- |
| Property Name | Description |
| Facility | The facility of the event that uses one of the values from the table that follows. |
| Severity | Numeric value that indicates the severity of the event using one of the following values:  0 - Emergency  1 - Alert  2 - Critical  3 - Error  4 - Warning  5 - Notice  6 - Info  7 - Debug |
| Priority | Numeric priority of the message. |
| PriorityName | Text description of the priority level. |
| TimeStamp | Time that the message was sent. |
| HostName | Name of the device sending the message. |
| Message | Text of the message |

Important

The event expression will almost always contain the Host Name in addition to one or more properties depending on the criteria that you require. Since a single management server may receive messages from multiple network devices, it must be able to determine which device sent a particular event. If the Host Name is not in the criteria, then a single event will most likely create a separate alert for each device.

Facility Values

The value for the facility property defines the part of the system that the message originated from. It will have one of the values from the following table:

|  |  |  |
| --- | --- | --- |
| Facility | Description | Value |
| 0 | Kernel | Kernel messages |
| 1 | User | User-level messages |
| 2 | Mail | Mail System |
| 3 | Daemons | System daemons |
| 4 | Auth | Security and authorization |
| 5 | Syslog | Syslog internal messages |
| 6 | LPR | Line printer subsystem |
| 7 | News | Network news |
| 8 | UUCP | Unix-to-Unix copy program |
| 9 | Cron | Cron daemon |
| 10 | Auth2 | Security and authorization |
| 11 | FTP | FTP daemon |
| 12 | NTP | Network time subsystem |
| 13 | LogAudit | Audit level |
| 14 | LogAlert | Message alert |
| 15 | Cron2 | Cron daemon |
| 16 | Local0 | Local use 0 |
| 17 | Local1 | Local use 1 |
| 18 | Local2 | Local use 2 |
| 19 | Local3 | Local use 3 |
| 20 | Local4 | Local use 4 |
| 21 | Local5 | Local use 5 |
| 22 | Local6 | Local use 6 |
| 23 | Local7 | Local use 7 |

Configure Alerts

The Configure Alerts page is only available for monitors and alerting rules. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating Syslog Event Rules

The following procedure shows how to create a Syslog event alerting rule in Operations Manager with the following details:

 Runs on all network devices.

 Generates an alert for any message with a severity of error or worse.

To create a Syslog event alerting rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then select Rules.  3. Right-click Rules and select Create a new rule.  4. On the Rule Type page, do the following:  a. Expand Alert Generating Rules, expand Event Based, and then click Syslog (Alert).  b. Select the management pack from step 1.  c. Click Next.  5. On the General page, do the following:  a. In the Rule Name box, type Alert on syslog message.  b. In the Rule Category box, select Alert.  c. Next to Rule Target click Select.  d. Select View all targets.  e. In the list of targets, select Node and then click OK.  f. Leave Rule is enabled selected.  g. Click Next.  6. On the Build Event Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type Severity.  c. In the Operator box select Less than or equal to.  d. In the Value box type 3.  e. Click Insert.  f. In the Parameter Name box type HostName.  g. In the Operator box select Equals.  h. Click the ellipse button next to Value and click SNMP Agent Address.  i. Click Next.  7. On the Configure Alerts page, do the following:  a. In the Alert name box, type Syslog error message received  b. In the Alert description box, type $Data/EventData/DataItem/Message$.  c. Click Create. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

SNMP Events

SNMP monitors and rules in Operations Manager allow you to retrieve messages from computers and devices that support Simple Network Management Protocol (SNMP). You can create rules and monitors that either wait for an SNMP trap to be sent or that retrieve information on a periodic basis using an SNMP probe.

Target

Rules and monitors run on the agent computer of each instance of the target class, and they usually access data on the local computer. SNMP rules and monitors typically work with information from a computer or device different from the one running the monitors or rules. For SNMP traps, the monitor or rule needs to be running on the agent that receives the trap. The device needs to be configured to deliver traps to this agent. For SNMP probes, the monitor or rule needs to be running on any agent that is authorized to access the device with SNMP. The device may need to be configured to allow communication from this agent.

Network devices that are discovered with the Discovery Wizard are managed by a resource pool that you specify during the discovery process. A resource pool contains one or more management servers. You can use the classes for these devices as targets, and the rule or monitor will run on each computer in the resource pool. In this case, the device will need to send SNMP traps to each of the computers in the pool and allow access to each computer in the pool for SNMP probes.

SNMP Event Wizards

The table below lists the wizards that are available for both simple and delimited text files.

|  |  |
| --- | --- |
| Management Pack Object | Wizards Available |
| Monitors | SNMP probe monitor with single event reset |
| SNMP trap monitor with single event reset |
| Rules | Alert Generating SNMP trap rule |
| Event collection SNMP probe rule |
| Event collection SNMP trap rule |
| Performance collection SNMP probe rule |

SNMP Event Wizard Options

When you run an SNMP monitor wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule or wizard including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule or monitor. For a rule, the name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. For a monitor, the name appears in the Health Explorer of any target objects. |
| Description | Optional description of the rule or monitor. |
| Management Pack | Management pack file to store the rule or monitor.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category (Rules only) | The category for the rule. For an event collection rule, this should be Event Collection. For a performance collection rule, this should be Performance Collection. For an alerting rule, this should be Alert. |
| Parent Monitor (Monitors only) | The aggregate monitor that the monitor will be positioned under in the Health Explorer. For more information, see [Aggregate Monitors](#ze0b5ecea287f4ec789f9235000adb22a). |
| Target | The class to use for the target of the rule or monitor. The rule or monitor will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f).  If you are monitoring a network device discovered in the Discovery Wizard, then use the class for the device or one of its components, depending on what the monitor most applies to. |
| Rule is enabled  Monitor is enabled | Specifies whether the rule or monitor is enabled. |

SNMP Probe / SNMP Trap Provider

SNMP probe rules have an SNMP Probe page, while SNMP trap rules have an SNMP Trap Provider page. SNMP monitors will have two of the appropriate page, one to define the healthy state and the other to define the warning or critical state. The page defines the community string and OID of the SNMP probe or trap.

|  |  |
| --- | --- |
| Option | Description |
| Frequency (Probe only) | The frequency that the probe is run. A frequency that is configured too low can result in excess overhead on the device being monitored. A frequency that is configured too high can result in the monitor not detecting a problem quickly. A frequency from 2 minutes to 15 minutes is a common range. |
| Community string | If Use discovery community string is selected, then the community of the target device is used. If Use custom community string is selected, then you can specify a community string. |
| Object Identifier | For a probe, one or more Object Identifiers (OID) to retrieve from the device. A value for each one will be collected and available for evaluation in the expression. Most rules and monitors will use a single OID, but multiple OIDs can be used.  For a trap, one or more Object Identifiers (OID) to listen for from the device. Most rules and monitors will use a single OID, but multiple OIDs can be used. |
| All Traps (Trap only) | If select, the Object Identifier list is disabled, and all traps from the target object will be collected, regardless of the OID. |

Build Expression (Monitors Only)

SNMP monitors have a Build Expression page for each of the SNMP Probe or SNMP Trap Provider pages. The expression evaluates the SNMP data returned to determine the health state of the monitor.

For more information about expressions, [Expressions](#z5e4020ce75c54c178719baeeafd546a5).

The Parameter Name in each expression requires a variable referring to a piece of data from the SNMP probe or trap. The data that is available includes header information and a data element for each OID specified. The header information is shown in the following table:

|  |  |
| --- | --- |
| Data Item | Description |
| Source | IP address of the device. |
| Destination | IP address of the agent receiving the event |
| CommunityString | Encrypted community string |
| ErrorCode | Error code returned by the request |
| Version | Version of SNMP used |

The information in each data element is shown in the following table:

|  |  |
| --- | --- |
| Data Item | Description |
| OID | OID of the data element |
| Syntax | Indicates the success or failure of the SNMP operation. If successful, the property is set to a value indicating the data type of the value. If unsuccessful, the property is set to a data type indicating the error. The specific values are listed in the documentation for the [SNMP Probe Module](http://go.microsoft.com/fwlink/?LinkID=230974). |
| Value | The value of the data element. |

To refer to the OID data elements, you can use the following syntax:

|  |  |  |
| --- | --- | --- |
| Syntax | Example | Description |
| SnmpVarBinds/SnmpVarBind/<ElementName> | SnmpVarBinds/SnmpVarBind/Value | Use this syntax when a single OID is used. |
| SnmpVarBinds/SnmpVarBind[#]/<ElementName> | SnmpVarBinds/SnmpVarBind[2]/Value | Use this syntax when you have multiple OIDs and want to refer to each by its numeric order. The first OID is 1, the second is 2, and so on. |
| SnmpVarBinds/SnmpVarBind[OID="<OID>"]/<ElementName> | SnmpVarBinds/SnmpVarBind[OID="1.3.6.1.2.1.1.5.0"]/Value | Use this syntax when you have multiple OIDs and want to refer to each by the specific OID. |

Configure Health

The Configure Health page is only available for monitors. It allows you to specify the health state that will be set for each of the events. The first event will typically set the monitor to Warning or Critical while the second event or the timer will set the monitor to Healthy.

Configure Alerts

The Configure Alerts page is only available for monitors and alerting rules. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating SNMP Monitors and Rules

Creating an SNMP Rule

Use the following procedure to create an SNMP performance collection rule in Operations Manager with the following details:

 Runs on all network devices by using Node for the target.

 Collects the number of open TCP connections (OID 1.3.6.1.2.1.6.9.0) every 10 minutes.

To create an SNMP Performance Collection Rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then select Rules.  3. Right-click Rules and select Create a new rule.  4. On the Rule Type page, do the following:  a. Expand Collection Rules, expand Performance Based, and then click SNMP Performance.  b. Select the management pack from step 1.  c. Click Next.  5. On the General page, do the following:  a. In the Rule Name box, type Collect Open TCP Connections.  b. In the Rule Category box, select Performance Collection.  c. Next to Rule Target click Select and then select Node.  d. Leave Rule is enabled selected.  e. Click Next.  6. On the SNMP Probe page, do the following:  a. In the Frequency box, 10 minutes.  b. In the Object Identifier box, type 1.3.6.1.2.1.6.9.0 and press ENTER.  c. Click Create. |

Creating an SNMP Monitor

Use the following procedure to create an SNMP trap monitor Operations Manager with the following details:

 Runs on all network devices by using Node for the target.

 Monitors for the status of a port. Link down is indicated with OID .1.3.6.1.6.3.1.1.5.3. Link up is indicated with OID .1.3.6.1.6.3.1.1.5.4.

 Monitors port 16 only. This is indicated by Object Identifier .1.3.6.1.2.1.2.2.1.8.16 with a value of 2 for link down and Object Identifier .1.3.6.1.2.1.2.2.1.8.16 with a value of 1 for link up.

 Includes the OID and value for the first four entries in the SNMP data.

To create an SNMP Trap monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace.  3. Right-click Monitors, select Create a Monitor, and then select Unit Monitor.  4. On the Monitor Type page, do the following:  a. Expand SNMP, then Trap Based Detection, then Simple Trap Detection, and then SNMP Trap Monitor.  b. Select the management pack from step 1.  c. Click Next.  5. On the General page, do the following:  a. In the Name box, type Port active.  b. Click Select next to the Monitor Target box.  c. Select Node and click OK.  d. In the Parent Monitor box, select Availability.  e. Leave the Monitor is enabled box checked, select and click Next.  6. On the First SnmpTrapProvider page, do the following:  a. In the Object Identifier box, type .1.3.6.1.6.3.1.1.5.3 and press ENTER.  b. Click Create.  7. On the Build First Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type SnmpVarBinds/SnmpVarBind[OID=".1.3.6.1.2.1.2.2.1.8.16"]/Value.  c. In the Operator box select Equals.  d. In the Value box type 2.  e. Click Next.  8. On the Second SnmpTrapProvider page, do the following:  a. In the Object Identifier box, type .1.3.6.1.6.3.1.1.5.4 and press ENTER.  b. Click Create.  9. On the Build Second Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type SnmpVarBinds/SnmpVarBind[OID=".1.3.6.1.2.1.2.2.1.8.16"]/Value.  c. In the Operator box select Equals.  d. In the Value box type 1.  e. Click Next.  10. On the Configure Health page, do the following:  a. Next to FirstEventRaised, change the Health State to Critical.  b. Click Next.  11. On the Configure Alerts page, do the following:  a. Check Generate alerts for this monitor  b. In the Generate an alert when box, select The monitor is in a critical health state.  c. Leave the box selected to automatically resolve the alert.  d. In the Alert name box, type Port active  e. Click the ellipse button next to the Alert description box.  f. Clear the contents of the Value box.  g. Click Data and then Source. Press ENTER.  h. Click Data and then Destination. Press ENTER.  i. Click Data, then SnmpVarBinds, then SnmpVarBind, and then OID.  j. In the variable, change [<<INT>>] to [1].  k. Type a space after the variable.  l. Click Data, then SnmpVarBinds, then SnmpVarBind, and then Value.  m. In the variable, change [<<INT>>] to [1].  n. Repeat the previous steps to add the OID and value for entries 2, 3, and 4.  o. Click OK.  12. Click Create. |

See Also

Monitoring Networks by Using Operations Manager 2012

How to Discover Network Devices in Operations Manager 2012

Event Monitor Logic

Event monitors in System Center 2012 – Operations Manager use one of the event data sources to identify a particular event that indicates an issue. As soon as the specific data source that holds the required information is identified, the logic used to determine different health states must be determined. In addition to the logic that indicates whether an error condition has occurred, additional logic must be defined to determine when the state should be changed back to a healthy condition.

The different kinds of logic that can be used to detect an error condition by using events are listed in the following table. See the individual topic for each condition for details on how its logic is implemented. As noted in the table, some logic can only be used with Windows events.

|  |  |  |
| --- | --- | --- |
| Logic | Data Sources | Description |
| Simple Event | All | Detects an error state from the occurrence of a single event. There is no individual topic for this logic. |
| [Repeating Events](#z9dacca1d43be4caaad2a2183c14a0286) | All | Detects an error state from one or more occurrences of a particular event in a specified time window. |
| [Correlated Events](#ze164687325154876b03a353e1735c0b4) | Windows Events | Detects an error state from the occurrence of two events in a specified time window. |
| [Correlated Missing Events](#z71cbd5d7d84e4e1bac8e35005890a84a) | Windows Events | Detects an error state from an expected event not being detected in a particular time window after the occurrence of another event. |
| [Missing Events](#zc8a08de42f184fd8a2eba151b4488c66) | Windows Events | Detects an error state from an expected event not being detected in a particular time window. |

Repeating Events

Repeated event detection in Operations Manager uses one or more occurrences of a particular event in a time window to indicate an error condition in a monitor. Repeated event logic is available for the following monitors:

 Windows Events

 Text Logs

 WMI Events

This typically applies to conditions in an application where a single event on its own can be ignored, but multiple occurrences of that event in a particular time window indicate a potential error. There are different algorithms that can be used for this detection, depending on the logic that best identifies the specific application issue. The following are details of the different algorithms:

Trigger on Timer

Trigger on timer consolidation of events uses a specified time window and is not dependent on the number of events received. A single event can trigger an error in the health state as in simple detection. Unlike simple detection which sets the health state immediately upon detection of the specified event, however trigger on timer consolidation waits until a specified time window to set the health state of the monitor. The time window can be a rotating time duration of specified length or a specific window based on day of the week.

Trigger on timer consolidation is useful for errors that should only be detected in a certain time window. Used with a time window based on a specific time of day, this disables the monitor outside that time period. It can also have the effect of delaying the change of state for a particular time during which an event that indicates a healthy state could be received. In this case, the health state would never be changed.

Trigger on Count

Trigger on count consolidation of events lets a monitor require multiple occurrences of the same event in a specified time window before it changes the health state to an error. The time window can be rotating time duration of specified length or a specific window based on day of the week.

Trigger on count consolidation resembles trigger on timer consolidation except that multiple occurrences of the event are required instead of just one. When the time window is reached, the event count is returned to zero, and the specific number of events must detected before the time window expires again for the health state to be changed.

Trigger on Count, Sliding

Trigger on count, sliding consolidation of events is similar to trigger on count consolidation except that the time window is reset every time that the specified event is received. The time window only expires if the time is reached after the occurrence of the last event.

Trigger on count, sliding consolidation is useful for error conditions that are detected by a certain number of events in a particular length of time. By using trigger on count consolidation, some events could be received in one time window and then other events received in the next time window with the result that the health state is never changed. Using trigger on count, sliding consolidation, the time window depends on when the event occurs preventing this condition.

Repeated Events Example

To help with understanding the different algorithms used for repeated event detection, the following table shows the effect on health state for monitors based on the different kinds of consolidation. This is based on a repeated event monitor that uses the following details:

 Consolidation interval: 2 minutes

 Compare count: 3 (ignored by Trigger on Timer)

 Health state on repeated event: Critical

 Reset Logic: Event reset using Event 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time | Event | Trigger on Timer | Trigger on Count | Trigger on Count, Sliding |
| 00:00:00 | - | Healthy | Healthy | Healthy |
| 00:01:00 | Event 1 | Healthy | Healthy | Healthy |
| 00:02:00 | - | Healthy | Healthy | Healthy |
| 00:02:30 | - | Healthy | Healthy | Healthy |
| 00:03:00 | - | Critical | Healthy | Healthy |
| 00:03:30 | Event 3 | Healthy | Healthy | Healthy |
| 00:04:00 | Event 1 | Healthy | Healthy | Healthy |
| 00:04:30 | - | Healthy | Healthy | Healthy |
| 00:05:00 | Event 1 | Critical | Healthy | Healthy |
| 00:05:30 | - | Critical | Healthy | Healthy |
| 00:06:00 | - | Critical | Healthy | Healthy |
| 06:30:00 | Event 1 | Critical | Healthy | Healthy |
| 07:00:00 | Event 1 | Critical | Healthy | Critical |
| 07:30:00 | - | Critical | Healthy | Critical |
| 00:08:00 | Event 1 | Critical | Healthy | Critical |
| 00:08:30 | - | Critical | Critical | Critical |
| 00:09:00 | Healthy | Critical | Healthy | Healthy |

 Using trigger on timer, a critical state is set at 00:03:00 event though the event is received at 00:01:00 because the time window starts when the monitor is loaded. The start is reset to healthy at 00:03:30, but the critical state is again triggered at 00:05:00 from the time window started at 00:03:00.

 Using trigger on count, the event at 00:05:00 does not trigger a critical state because the time window started by the event at 00:01:00 would have expired at 00:03:00. This event is instead part of the time window started by the event at 00:04:00 which expires at 00:06:00. The monitor triggers a critical state at 00:08:30 because of the 3 events detected in the time window started with the event at 00:06:30.

 Using trigger on count, sliding, each occurrence of Event 1 starts its own window. The critical state is triggered at 00:07:00 from the 3 events detected in the time window started with the event at 00:05:00.

Creating Repeated Event Monitors

The following procedure describes how to create a repeating event monitor with the following details:

 Positioned under the Availability aggregate monitor.

 Sets the monitor to a critical state when three events with source EventCreate and number 201 are detected in the Application event log within 10 minutes.

 Resets after fifteen minutes of no event being received.

 Creates an alert when the monitor enters a critical state.

Note

EventCreate is used as the event source so that the EventCreate utility can be used to create a test event. This utility is available on any Windows Computer and creates test events with a source of EventCreate. If you have another method of creating test events, then you can use a different source.

To create a repeating event monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace.  4. Select Management Pack Objects.  5. Right-click Monitors, select Create and Monitor, and then select Unit Monitor.  6. On the Monitor Type page, do the following:  a. In the Select the type of monitor to create box, expand Windows Events and then Repeated Event Detection.  b. Select Timer Reset.  c. In the Management Pack dropdown list, select the management pack for the application.  d. Click Next.  7. On the General page, do the following:  a. In the Name box, type Repeated error event 201 or another name for the monitor. This is the text that will appear in the Health Explorer.  b. Click Select.  c. In the Select Items to Target dialog box, select the name that you used for the Windows Service template in step 2.  d. The Parent monitor box should show Availability. You can select a different parent monitor.  e. Ensure that Availability is selected for the Parent monitor.  f. The Monitor is enabled box should be checked so that the monitor is enabled.  g. Click Next.  8. On the Event Log Name page, do the following:  a. In the Log Name box, keep the default value of Application.  b. Click Next  9. On the Event Expression page, do the following:  a. For the Event ID value, type 201  b. For the Event Source value, type EventCreate  c. Click Next.  10. On the Repeated Settings page, do the following:  a. For the Counting Mode, select Trigger on count, sliding.  b. For the Compare Count, type 3.  c. Select Based on items occurrence within a time interval.  d. For the Interval, type 10 and select Minutes.  e. Click Next.  11. On the Auto Reset Timer page, do the following:  a. For Specify wait time, select 15 minutes.  b. Click Next  12. On the Configure Health page, do the following:  a. For Timer Event Raised, leave the Health State as Healthy.  b. For Repeated Event Raised, set the Health State to Critical.  c. Click Next.  13. On the Configure Alerts page, do the following:  a. Check Generate alerts for this monitor.  b. For Generate an alert when: leave the default of The monitor is in a critical health state.  c. Leave the box Automatically resolve the alert when the monitor returns to a health state checked.  d. For the Alert name, leave the default which is the name of the monitor or replace it with different text. This will be the name of the alert that appears in the Operations console when the alert is created.  e. Leave the default Priority of Medium.  f. Leave the default Severity of Critical. Note that you can change the alert severity to Warning even though the monitor is set to Critical  g. In the Alert description box, type Event 201 was detected $Data/Context/Count$ times between $Data/Context/TimeWindowStart$ and $Data/Context/TimeWindowEnd$. The first event was at $Data/Context/TimeWindowStart$. The last event was at $Data/Context/TimeWindowEnd$.. Rather than typing in each of the $Data variables, you can select them by clicking the ellipse button and then selecting Data and the property.  h. Click Create. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

[Text Logs](#za3d2ce59a94b4dd08aeb5f985c56a964)

[WMI Events](#z3474a6f4bca94492a6534459e19b1d2a)

Correlated Events

A correlated event monitor in Operations Manager uses two separate events in a particular time period to detect a single issue. This kind of monitor supports conditions where an issue cannot be identified by a single event alone.

When the first event is detected, a timer is triggered. If the second event is received within that period, the state change is triggered. If the second event is not received in the period, the timer is reset until the first event is received again. The monitor may be configured to better tune the specific conditions that must be met in order to perform correlation. These options include the following:

 Whether the events must be in chronological order. One of the events may always be expected before the other one, or they may be expected in either order.

 Whether the first or last occurrence of the first event should be used. If the first occurrence is specified, then each occurrence of the first event will have its own time window and search for corresponding occurrences of the second event. With the last occurrence specified, if the first event reoccurs with the time window, then the time window is extended based on the last event. The monitor can also be configured to reset the time window every time that the first event occurs. When the time window is reset, all previous occurrences of both events are ignored.

 The number of occurrences of the second event that must be received to trigger the state change. Instead of changing the health state after receiving a single instance of the two events, multiple instances of the second event may be required.

 Properties between the first and second event that must match for correlation to be performed. Instead of detecting two occurrences of each event, additional comparison may be required to determine whether the events are related. The monitor can, for example, confirm that a particular parameter matches between the two events to make sure that they match.

Correlated Events Example

The following table provides an example of a correlated event monitor by using the first and the last occurrence of the first event. The monitor uses the following details:

 Event Log A: Event 1

 Event Log B: Event 2

 Correlation interval: 2 minutes

 Number of occurrences of Event 2: 3

 Health state on correlation: Critical

 Reset Logic: Event reset using Event 3

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Event | First Occurrence | Last Occurrence |
| 00:00:00 | - | Healthy | Healthy |
| 00:01:00 | Event 1 | Healthy | Healthy |
| 01:30 | Event 2 | Healthy | Healthy |
| 00:02:00 | Event 2 | Healthy | Healthy |
| 00:02:30 | - | Healthy | Healthy |
| 00:03:00 | Event 1 | Healthy | Healthy |
| 00:03:30 | Event 2 | Healthy | Healthy |
| 00:04:00 | Event 2 | Healthy | Healthy |
| 00:04:30 | Event 1 | Healthy | Healthy |
| 00:05:00 | Event 2 | Critical | Healthy |
| 05:30:00 | Event 3 | Healthy | Healthy |
| 06:00:00 | Event 1 | Healthy | Healthy |
| 06:30:00 | Event 2 | Healthy | Healthy |
| 07:00:00 | Event 1 | Healthy | Healthy |
| 07:30:00 | Event 2 | Healthy | Healthy |
| 08:00:00 | Event 2 | Critical | Healthy |
| 08:30:00 | Event 2 | Critical | Critical |
| 09:00:00 | Event 3 | Healthy | Healthy |

 The First Occurrence does not trigger a critical state when Event 2 is detected at 00:03:00 because the timer was reset at 00:03:00 which is 2 minutes after the first occurrence of Event 1 at 00:01:00.

 The First Occurrence triggers a critical state at 00:05:00 because Event 2 is detected 3 times within the 2 minutes since the first occurrence of Event 1 at 00:03:00. Event 1 starts a new time window at 00:03:00 because the time window from Event 1 at 00:01:00 would have expired.

 The First Occurrence triggers a critical state at 00:08:00 because Event 2 is detected 3 times within 2 minutes from Event 1 at 00:06:00.

 The First Occurrence resets its state to healthy at 00:05:30 and 00:09:00 because Event 3 is detected.

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

Missing Events

Instead of detecting a particular event to identify an error condition, a missing event monitor in Operations Manager uses the absence of a particular event in a particular time window to determine an error. This supports applications that are expected to generate an informational event that indicates a successful operation or the success of a particular action.

For example, consider an application that performs a scheduled data transfer each evening and creates an event when it has completed successfully. A missing event monitor could be created that searches for the event in a particular time window each evening. If the event is detected, then the monitor remains in a healthy state. If it is not found, then it enters error state that indicates that the last transfer did not occur successfully.

Missing Event Example

The following table provides an example of a missing event monitor by using the following details:

 Event: Event 1

 Fixed Schedule: Su-Sa 2:00 AM – 3:00 AM

 Health state on missing event: Critical

 Reset Logic: Event reset using Event 3

|  |  |  |
| --- | --- | --- |
| Time | Event | Health State |
| 00:00:00 | - | Healthy |
| 00:01:00 | Event 1 | Healthy |
| 00:02:00 | - | Healthy |
| 00:03:00 | - | Critical |
| 00:04:00 | - | Critical |
| 00:05:00 | Event 3 | Healthy |

 The critical state is triggered at 00:03:00 when Event 1 is not detected within the specified window.

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

Correlated Missing Events

A correlated missing event monitor in Operations Manager determines an error by the absence of a particular event after the occurrence of another. This resembles the missing event monitor except that instead of searching for the missing event in a particular time window, the monitor searches for the event in a particular time after another event is first detected.

For example, consider an application that performs a backup each evening and creates an event when it starts and a second event when it has completed successfully. A correlated missing event monitor could be created that searches for the event in a particular time window each evening. If both events are detected, then the monitor remains in a healthy state. If the first is found, then the timer starts. If the time is reached before the second event is detected, then the state change is triggered to indicate that the last backup did not occur successfully.

Correlated Missing Events Example

The following table provides an example of a correlated missing event monitor by using the first and the last occurrence of the first event. The monitor uses the following details:

 Missing Event Log A: Event 1

 Missing Event Log B: Event 2

 Correlation interval: 2 minutes

 Number of occurrences of Event 2: 3

 Health state on correlation: Critical

 Reset Logic: Event reset using Event 3

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Event | First Occurrence | Last Occurrence |
| 00:00:00 | - | Healthy | Healthy |
| 00:01:00 | Event 1 | Healthy | Healthy |
| 1:30 | Event 2 | Healthy | Healthy |
| 00:02:00 | Event 2 | Healthy | Healthy |
| 00:02:30 | Event 1 | Healthy | Healthy |
| 00:03:00 | - | Critical | Healthy |
| 00:03:30 | Event 2 | Critical | Healthy |
| 00:04:00 | Event 2 | Critical | Healthy |
| 00:04:30 | - | Critical | Critical |
| 00:05:00 | Event 3 | Healthy | Healthy |

 The First Occurrence triggers a critical state at 00:03:00 because Event 2 has not been detected 3 times in the 2 minute interval since the first occurrence of Event 1 at 00:01:00.

 The Last Occurrence does not trigger a critical state at 00:03:00 because Event 1 occurs at 00:02:30 resetting the timer. The critical state is not triggered until 00:04:30 when Event 2 has not been detected in the 2 minutes interval since the last occurrence of Event 1 at 00:02:30.

 The single occurrence of Event 3 at 00:05:00 resets both monitors to healthy.

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

Event Monitor Reset

Unlike other kinds of monitors in System Center 2012 – Operations Manager, it may be difficult to define the criteria to return an event monitor to a healthy state. This is because applications often generate an event when a problem occurs but do not create a corresponding event when the problem has been corrected. For this reason, you have the following options for setting the healthy state for an event monitor.

|  |  |
| --- | --- |
| Reset Logic | Description |
| Event Reset | A single specific event indicates that monitor should be reset. |
| Manual Reset | The monitor is never automatically reset. The user must manually reset the monitor. |
| Timer Reset | The monitor is automatically reset after a specified time. |

Each of these methods is discussed at length in the following sections:

Event Reset

With event reset, the monitor is reset when a single occurrence of a specific event is detected. The event must be the same type as the event used for detecting the error condition. For example, a Windows event monitor might specify an event with a particular event source and number to indicate an error condition. Another Windows event with the same event source but a different number might indicate that the error in the application was corrected.

Event reset can only be used if the application provides an event indicating the particular error was corrected. Many applications create an event when an error occurs but may not create a corresponding event that indicates that the error was corrected. Event reset cannot be used in this case.

Manual reset

With manual reset, the monitor never returns to a healthy state automatically. The user must determine whether the problem was corrected and then select the monitor in the Health Explorer and select Reset Health.

The advantage to this strategy is that a monitor can be used for issues that do not create an event that indicates a healthy state. The monitor can affect the health state of the managed object instead of creating a simple alert from a rule. The downtime will be recorded for the object in the State Change Events in the Operations Console and in any availability reports.

There are multiple implications of this strategy that should be considered. The first is the additional work required from the user because the monitor will never automatically reset. It can also result in too much downtime being recorded if the user waits a long time before performing the reset. The problem may have been corrected fairly quickly, but the healthy state will not be recorded until the user performs the reset.

Use of manual reset should be especially cautioned for monitors where there is a potential for a single problem to affect multiple instances of the target class. Because users cannot reset the monitor for multiple instances in the Operations Console, the user would be required to manually open the Health Explorer for each instance to perform this action. Depending on the number of instances, this could result in significant effort for the user.

Timer Reset

A timer reset acts the same as a manual reset except that if the user does not manually reset the monitor after a specified time, it will reset automatically. One use of this kind of reset is for issues that continuously log error events until the problem is corrected. Instead of using another event to indicate that the problem was corrected, the previously detected error event for a specified period can be used as the success criteria.

The timer reset can be used in the place of a manual reset providing the advantage of automatically resetting after a while if the user does not perform a manual reset.

Which reset should I use?

 If the application you’re monitoring creates an event when the problem has been corrected, use Event Reset. This is the preferred method since the monitor will return to a healthy state as soon as soon as is appropriate without any user intervention. Any alert generated from the monitor can also be closed automatically.

 If the application you’re monitoring does not create an event when the problem has been corrected, then you should use Manual Reset or Timer Reset. Refer to the descriptions of each above to determine which strategy is most appropriate for your specific purposes.

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Windows Events](#z836cd87dc9fe4b8e86c4f8bdd23cc995)

[Text Logs](#za3d2ce59a94b4dd08aeb5f985c56a964)

UNIX/Linux Shell Command Alerts

UNIX/Linux shell commands can be used to detect events and generate alerts. When the rule is run, the provided command is executed on the agent, and if the output matches the provided filter, the alert is generated.

Target

Rules and monitors run on the agent computer of each instance of the target class, and they usually access data on the local computer. The target must be a UNIX and Linux computer type, such as UNIX/Linux Computer, Linux Computer, etc.

UNIX/Linux Shell Command (Alert) Wizard Options

When you run a UNIX/Linux Shell Command rule wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

Rule Type

The Rule Type page includes basic settings for the rule including its type and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Select the type of rule to create | To create an alert-generating rule based on the execution of an UNIX/Linux shell command, select UNIX/Linux Shell Command (Alert). |
| Management Pack | Management pack file to store the rule or monitor.For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the rule. The name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. |
| Description | Optional description of the rule. |
| Rule Category | The category for the rule. For a performance collection rule, this should be Performance Collection. For an alerting rule, this should be Alert. |
| Rule Target | The class to use for the target of the rule. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is enabled | If checked, the rule is enabled and the shell command will run according to the schedule. If unchecked, the rule is not enabled and the script will not run. The rule can be enabled for a group of target objects by creating an override to enable the rule. |

Schedule

The following options are available on the Schedule page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Run every | Frequency that the script should be run. This should typically not be less than 5 minutes. |
| Synchronize at | If enabled, the schedule will be synchronized to occur at the specified time. |

UNIX/Linux Shell Command

The following options are available on the Shell Command Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Command | The shell command to execute. This can be the full path to a program or script, a command, or a one-line sequence of multiple commands (using pipeline operators). |
| Run As Profile | Either the “UNIX/Linux Action Account” or “UNIX/Linux Privileged Account” profile. Select the profile that associates the required account credentials with the task target. The associated account will be used to execute the command. |
| Timeout (seconds) | The number of seconds that the command can run before the agent stops it. This prevents problem commands from running continuously and putting excess overhead on the agent computer. |

Filter Expression

The Filter Expression page allows you to filter for output to generate an alert. The alert is generated only if the output of the shell command matches the filter expression.

|  |  |
| --- | --- |
| Property Name | Description |
| Filter one or more events | An expression that filters output of the shell command. For more information on building expressions see [Expressions](#z5e4020ce75c54c178719baeeafd546a5) The Parameter Name syntax for command execution output is: StdOut: //\*[local-name()="StdOut"]StdErr: //\*[local-name()="StdErr"]Return Code: //\*[local-name()="ReturnCode"] |

Configure Alerts

The Configure Alerts page is used to define alert properties for the rule. Its options are explained in [Alerts](#z118b16cc01c1462d8d10c2064ba51db0).

Creating UNIX/Linux Shell Command (Alert) Rules

The following procedure shows how to create an UNIX/Linux shell command alerting rule in Operations Manager with the following details:

1. Runs on all UNIX/Linux Computers every 15 minutes

2. Generates an alert if the file “/tmp/error” exists

To create a UNIX/Linux shell command alerting rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then select Rules.  3. Right-click Rules and select Create a new rule.  4. On the Rule Type page, do the following:  a. Expand Alert Generating Rules, expand Event Based, and then click UNIX/Linux Shell Command (Alert).  b. Select the management pack from step 1.  c. Click Next.  5. On the General page, do the following:  a. In the Rule Name box, type Alert on Error File Exists.  b. In the Rule Category box, select Alert.  c. Next to Rule Target click Select and then select UNIX/Linux Computer.  d. Leave Rule is enabled selected.  e. Click Next.  6. On the Schedule page, do the following:  a. In the Run Every boxes, input 15 and Minutes.  b. Click Next.  7. On the Shell Command Details page, do the following:  a. In the Command box, type ls /tmp/error | wc –l. This command sequence will return a 1 if the file “/tmp/error” exists, and a 0 if it does not.  b. In the Run As Profile box, select the UNIX/Linux Action Account profile.  c. In the Timeout (Seconds) box, input 120.  d. Click Next.  8. On the Filter Expression page, do the following:  a. Configure an And expression with the entries:  i. //\*[local-name()=”StdOut”] equals 1  ii. //\*[local-name()=”ReturnCode”] equals 0  b. This will trigger an alert whenever the value of the shell command output is 1, and the command executed successfully.  9. On the Configure Alerts page, do the following:  a. In the Alert name box, type Error File Found.  b. In the Alert description box, type The file /tmp/error was found on the computer:  c. Click the […] button.  d. Click Target and select Network Name.  e. Click OK.  f. Click Create. |

See Also

[Event Monitors and Rules](#zab45f9efbdb44590a5c65a1a3d70ce9e)

[Alerts](#z118b16cc01c1462d8d10c2064ba51db0)

Performance Monitors and Rules

Performance counters are numeric data that is used to measure the performance of some aspect of the application. Performance data usually comes from Windows performance counters but can also come from WMI and from scripts. Measuring and collecting performance data with scripts is covered in [Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28).

Performance Monitors and Rules Topics

Details on each performance data source are provided in the following sections. This includes the information that is required to retrieve the required data, properties available in the resulting data, and what workflows the performance data source supports.

 [Performance Monitors](#z97cb90eebbaf4b5f8ca7094c89c5e7ba)

Details on the different types of calculations that can be performed for performance monitors.

 [Windows Performance Collection Rules](#z5f5058960954403dbdc819e372212c6c)

Rules and monitors using Windows performance counters which are sampled at specified intervals.

 [WMI Performance](#zc078fa8e264440499efc0ee1e589d5ab)

Rules and monitors that run a WMI query at specified interval and uses the value of a numeric property for performance data.

 [UNIX/Linux Shell Command Performance Collection Rules](#zd9c1f2f11b69413da558c1c45d9e34dd)

Rules that execute a UNIX/Linux command, script, or one-line sequence of multiple commands (using pipeline operators) and collect the output as numerical performance data.

Performance Monitors

Multiple kinds of calculations may be performed to determine the threshold for a performance monitor in Operations Manager. These threshold types are listed in the following table:

|  |  |  |
| --- | --- | --- |
| Threshold Type | Number of States | Description |
| [Average Threshold](#z71) | 2 | Compare the average of multiple collected values to a threshold. |
| [Consecutive Samples](#z72) | 2 | Compare several consecutive values to a threshold. All collected values must match the threshold criteria. |
| [Delta Threshold](#z73) | 2 | Compare the change between two consecutive values to a threshold. |
| [Double Threshold](#z74) | 3 | Compare a single collected value to two thresholds with one that indicates a Warning state and the other that indicates a Critical state. |
| [Simple Threshold](#z75) | 2 | Compare a single collected value to a threshold. |

Each kind of logic is described in detail in the following sections:

Simple Threshold

The simple threshold type is the most basic kind of performance threshold. A single numeric value is provided for the threshold. This threshold is compared to the measured value of the performance data.

Simple threshold supports a two state monitor. One state is set by a performance value equal to or less than the threshold. The other state is set by a performance value greater than the threshold.

Double Threshold

The double threshold type is similar to the simple threshold type but allows for two thresholds to be specified. Each threshold is compared to the measured value of the performance data.

Double threshold supports a three state monitor. One state is set by a performance value less than the low threshold. Another state is set by a performance value that is greater than or equal to the low threshold or one that is less than or equal to the high threshold. Another state is set by a value that is greater than the high threshold.

The following table provides an example of a double monitor by using the following details:

 Sample rate: 5 minutes

 Low threshold value: 10

 High threshold value: 15

 Over Upper Threshold State: Critical

 Between Thresholds State: Warning

 Under Lower Threshold State: Healthy

|  |  |  |
| --- | --- | --- |
| Time | Value | State |
| 00:00:00 | 5 | Healthy |
| 00:05:00 | 10 | Warning |
| 00:10:00 | 12 | Warning |
| 00:15:00 | 9 | Healthy |
| 00:20:00 | 12 | Warning |
| 00:25:00 | 16 | Critical |
| 00:30:00 | 15 | Critical |
| 00:35:00 | 8 | Healthy |

 The warning threshold is first exceeded at 00:05:00, but the value does not exceed the critical threshold.

 The critical threshold is first exceeded at 00:25:00 when the state is changed from warning to critical.

 The state is returned to a healthy state at 00:15:00 and 00:35:00 when the sampled value is less than the warning threshold.

Average Threshold

The average threshold type calculates the average of a specified number of consecutive samples and compares it to the specified threshold.

Average threshold supports a two state monitor. One state is set by an average performance value equal to or less than the threshold. The other state is set by an average performance value greater than the threshold.

The following table provides an example of an average threshold monitor by using the following details:

 Sample rate: 5 minutes

 Threshold value: 10

 Number of samples: 3

 Over Threshold State: Critical

 Under Threshold State: Healthy

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Value | Average | State |
| 00:00:00 | 5 | - | Healthy |
| 00:05:00 | 10 | - | Healthy |
| 00:10:00 | 12 | 9.0 | Healthy |
| 00:15:00 | 9 | 10.3 | Critical |
| 00:20:00 | 12 | 11.0 | Critical |
| 00:25:00 | 14 | 11.7 | Critical |
| 00:30:00 | 11 | 12.3 | Critical |
| 00:35:00 | 4 | 9.7 | Healthy |

 Because the specified number of samples for the average calculation is 3, no value is evaluated until the third sample.

 The value of 12 sampled at 00:10:00 exceeds the threshold value, but the calculated average from the last 3 samples is 9.0, which is under the threshold. The state is not changed.

 The value of 9 sampled at 00:15:00 does not exceed the threshold. But the calculated average from the last 3 samples is 10.3 which does exceed the threshold. The state is changed.

 The monitor does not return to a healthy state until 00:35:00 when the average from the last 3 samples drops the under the threshold value.

Consecutive Samples

The consecutive threshold type compares the threshold value to the performance counter for several consecutive samples. This supports monitors that should not be triggered by only a single value exceeding a threshold. The threshold must be exceeded multiple consecutive times to trigger a change in state.

Consecutive threshold supports a two state monitor. One state is set by the value being either greater than or less than the threshold value for each consecutive sample. The other state is set by a single sample not matching the other criteria.

The following table provides an example of a consecutive sample monitor by using the following details:

 Sample rate: 5 minutes

 Threshold value: greater than or equal to 10

 Number of samples: 3

 Over Threshold State: Critical

 Under Threshold State: Healthy

|  |  |  |
| --- | --- | --- |
| Time | Value | State |
| 00:00:00 | 5 | Healthy |
| 00:05:00 | 10 | Healthy |
| 00:10:00 | 12 | Healthy |
| 00:15:00 | 9 | Healthy |
| 00:20:00 | 12 | Healthy |
| 00:25:00 | 14 | Healthy |
| 00:30:00 | 11 | Critical |
| 00:35:00 | 8 | Healthy |

 The threshold is exceeded by the values sampled at 00:05:00 and 00:10:00, but the value at 00:15:00 is under threshold and resets the count.

 The value at 0:30:00 is the first time that 3 consecutive values have been sampled that exceed the threshold, so the state is changed.

 The single value at 00:35:00 is under the threshold and resets the monitor to a healthy state.

Delta Threshold

The delta threshold type compares the threshold value to the difference between two performance values. This might be two consecutive values or two values separated by a specified number of samples.

Delta threshold supports a two state monitor. One state is set by the difference of two values being greater than the threshold value. The other state is set by the difference of two samples being equal to or less than the threshold value.

The following table provides an example of a delta threshold monitor by using the following details:

 Sample rate: 5 minutes

 Threshold value: 10

 Number of samples: 3

 Over Threshold State: Critical

 Under Threshold State: Healthy

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Value | Delta |  |
| 00:00:00 | 7 | - | Healthy |
| 00:05:00 | 8 | - | Healthy |
| 00:10:00 | 13 | - | Healthy |
| 00:15:00 | 16 | 9 | Healthy |
| 00:20:00 | 21 | 13 | Critical |
| 00:25:00 | 24 | 11 | Critical |
| 00:30:00 | 25 | 9 | Healthy |

 Because the specified number of samples that the delta should be calculated from the current sampled value to the value 3 samples behind, no value is evaluated until the fourth sample.

 The delta calculation exceeds the threshold value at 00:20:00, and the state is changed.

 The monitor is reset at 00:30:00 when the delta calculation falls under the threshold.

To Create

Windows Performance Collection Rules

To define a collection rule in Operations Manager based on a Windows performance counter, the object name and counter name of the performance counter to sample must be defined with a frequency that specifies how frequently to sample the data. The instance name only has to be specified if the same counter will be collected for multiple objects on the same agent. If this is the case, a $Target variable will typically be used for the value in the instance name in order to differentiate between the performance values for different objects. The counter must be available on the agent computer that is running the rule or an error will be created in the Operations Manager event log on the agent.

Windows Performance Collection Wizard Options

When you run the Windows performance collection wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Rule Name | The name used for the rule. This appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. |
| Description | Optional description of the rule. |
| Management Pack | Management pack to store the rule.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category | The category for the rule. For a performance collection rule, this should be Performance Collection. |
| Rule target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |

Performance Counter

The Performance Counter page includes the definition of the performance counter to collect and the frequency it should be collected.

|  |  |
| --- | --- |
| Option | Description |
| Object | Text for the Object name. This is required. You can type in the name of the object or select a property from the target. |
| Counter | Name of the performance counter. |
| Instance | Text for the Instance name. This only required if the performance counter has multiple instances. You can type in the name of the instance or select a property from the target. |
| Include all instances for the selected counter | If select, the Instance box is disabled and the value for each instance of the performance counter is collected. |
| Interval | Specifies the frequency to collect the performance counter. |

Optimized Collection

The Optimized Collection page allows to you to enable and configure optimized collection for the counter. If you select optimization for a collection rule, a value is only collected if it differs from the previous sample by a specified tolerance, either an absolute value or a percentage. This helps reduce network traffic and the volume of data stored in the Operations Manager database. Optimization should be used for performance counters that are expected to only change gradually. For counters that are expected to very significantly from one value to the next, optimized collection should be disabled.

|  |  |
| --- | --- |
| Option | Description |
| Use Optimization | Specifies whether optimization should be enabled for the counter. If it is disabled, then every sampled value will be collected. |
| Absolute number | Specifies a value that the number must vary between the current sample and the previous sample for the value to be collected. The value can change in either positive or negative direction. |
| Percentage | Specifies a percentage of the previous sample that the difference between the current value and the previous value must be for the value to be collected. The change can be in either positive or negative direction. |

Creating Windows Performance Collection Rules

Use the following procedures to create a Windows performance collection rule in Operations Manager with the following details:

 Runs on all agents with a particular service installed.

 Collects the % Privileged Time for the selected service.

To create a Windows performance collection rule in Operations Manager

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Rules.  4. Right-click Rules and select Create a new rule.  5. On the Rule Type page, do the following:  a. Expand Collection Rules, expand Performance Based, and then click Windows Performance.  b. Select the management pack from step 1.  c. Click Next.  6. On the General page, do the following:  a. In the Rule name box, type % Privileged Time.  b. In the Rule Category box, select Performance Collection.  c. Next to Rule Target click Select and then select the name of the target that you created in step 2.  d. Leave Rule is enabled selected.  e. Click Next.  7. On the Performance Counter page, do the following:  a. Click Select.  b. In the Select Performance Counter dialog box, type a computer name or browse to a Computer that has the performance counter installed.  Note  The name of the computer is not recorded in the rule. The computer is only used to retrieve the details of the performance counter.  c. In the Object dropdown, select Process.  d. In Select counter from list, select % Privileged Time.  e. Click OK.  f. Clear the text in the Instance box.  g. Click the arrow to the right of the Instance box and select Service Name (Windows Service).  Note  You can also select the name of the service process when you select the counter. The strategy used here is to use the $Target variable to use the Service Name property of the target class. This will resolve to the name of the service when the rule runs. This is to illustrate the use of $Target variables.  h. Leave the Interval at its default value of 15 minutes.  i. Click Next.  8. On the Optimized Performance Collection Settings page, do one of the following:   Leave the Use Optimization option unselected.  Note  If you select optimization for a collection rule, a value is only collected if it differs from the previous sample by a specified tolerance, either an absolute value or a percentage. This helps reduce network traffic and the volume of data stored in the Operations Manager database. Optimization should be used for performance counters that are expected to only change gradually. In this example, the privileged time of the process is expected to vary significantly between samples so it would not benefit from optimized collection.   Click Create. |

See Also

[Performance Monitors and Rules](#z7d91fa3c13794325a87bbef3aa7def9d)

[Performance Monitors](#z97cb90eebbaf4b5f8ca7094c89c5e7ba)

WMI Performance

WMI performance refers to numeric data that is retrieved from a WMI query. This lets performance data be retrieved that is not available from a performance counter and without using the complexity and overhead of a script. The monitor or rule runs the query on a specified schedule and maps the value of the specified numeric field into the value property of the performance data.

For example, a monitor might have to track the size of a particular file. This might be a log file that indicates a particular problem when it exceeds a particular size. The name and size of the file could be retrieved from a query similar to the following:

Select Name, FileSize from CIM\_DataFile Where Name = 'C:\\MyApp\\MyAppLog.txt'

The monitor could run this query regularly by using the FileSize property for the value of the performance data and the Name property for the Instance property.

The WMI query returns a property bag with each property returned from the query. This set of properties will vary, depending on the class returned and the properties specified in the query. For more information about property bags, see [Property Bags](#z76).

Options

When you run the Windows performance collection wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Rule Name | The name used for the rule. This appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. |
| Description | Optional description of the rule. |
| Management Pack | Management pack to store the rule.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category | The category for the rule. For a performance collection rule, this should be Performance Collection. |
| Rule target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |

WMI Configuration

The Performance Counter page includes the definition of the performance counter to collect and the frequency it should be collected.

|  |  |
| --- | --- |
| Option | Description |
| WMI Namespace | The namespace containing the class used by the query. |
| Query | Name of the performance counter. |
| Query Interval | The frequency in seconds to run the query and collect the |

Performance Mapper

The Performance Mapper page is used to define values for the properties of the performance data being collected.

|  |  |
| --- | --- |
| Option | Description |
| Object | Text for the Object name. This is required. You can type in the name of the object or select a property from the target or from the property bag. |
| Counter | Text for the Counter name. This is required. You can type in the name of the object or select a property from the target or from the property bag. |
| Instance | Text for the Instance name. This only required if the target of the rule has multiple instances. You can type in the name of the object or select a property from the target or from the property bag. |
| Value | Numeric for the value for the performance. This will usually be a $Data variable to use a value from the property bag. |

Creating WMI Performance Collection Rules

Use the following procedure to create a WMI performance collection rule in Operations Manager with the following details:

 Runs on all agents with a particular service installed.

 Collects the size of a file called C:\MyApp\MyAppLog.txt.

To create a WMI performance collection rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Rules.  4. Right-click Rules and select Create a new rule.  5. On the Rule Type page, do the following:  a. Expand Collection Rules, expand Performance Based, and then click WMI Performance.  b. Select the management pack from step 1.  c. Click Next.  6. On the General page, do the following:  a. In the Rule name box, type Collect File Size with WMI.  b. In the Rule Category box, select Performance Collection.  c. Next to Rule Target click Select and then select the name of the target that you created in step 2.  d. Leave Rule is enabled selected.  e. Click Next.  7. On the WMI Configuration page, do the following:  a. In the WMI Namespace box, type root\cimv2.  b. In the Query box, paste the following WMI query.  Select Name,FileSize From CIM\_DataFile Where Name = 'C:\\Logs\\MyAppFile.txt'  c. In the Query Interval box, type 900.  d. Click Next.  8. On the Performance Mapper page, do the following:  a. In the Object box, type MyApplication.  b. In the Counter box, type File Size.  c. In the Instance box, type $Data/Property[@Name=’Name’]$.  d. In the Value box, type $Data/Property[@Name=’FileSize’]$.  e. Click Finish. |

See Also

[Performance Monitors and Rules](#z7d91fa3c13794325a87bbef3aa7def9d)

[Performance Monitors](#z97cb90eebbaf4b5f8ca7094c89c5e7ba)

UNIX/Linux Shell Command Performance Collection Rules

To define a collection rule in Operations Manager based on the output of an UNIX/Linux shell command, the command execution details, object name and counter name of the performance counter must be defined with a frequency that specifies how frequently to sample the data.

UNIX/Linux Shell Command Performance Collection Wizard Options

When you run the UNIX/Linux shell command performance collection wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

Rule Type

The Rule Type page includes basic settings for the rule including its type and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Select the type of rule to create | To create a performance collection rule based on the execution of an UNIX/Linux shell command, select UNIX/Linux Shell Command (Performance). |
| Management Pack | Management pack file to store the rule or monitor.For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Rule Name | The name used for the rule. This appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. |
| Description | Optional description of the rule. |
| Management Pack | Management pack to store the rule. For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category | The category for the rule. For a performance collection rule, this should be Performance Collection. |
| Rule Target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Rule is Enabled | If checked, the rule is enabled and the shell command will run according to the schedule. If unchecked, the rule is not enabled and the script will not run. The rule can be enabled for a group of target objects by creating an override to enable the rule. |

Schedule

The following options are available on the Schedule page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Run every | Frequency that the script should be run. This should typically not be less than 5 minutes. |
| Synchronize at | If enabled, the schedule will be synchronized to occur at the specified time. |

UNIX/Linux Shell Command

The following options are available on the Shell Command Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Command | The shell command to execute. This can be the full path to a program or script, a command, or a one-line sequence of multiple commands (using pipeline operators). |
| Run As Profile | Either the “UNIX/Linux Action Account” or “UNIX/Linux Privileged Account” profile. Select the profile that associates the required account credentials with the task target. The associated account will be used to execute the command. |
| Timeout (seconds) | The number of seconds that the command can run before the agent stops it. This prevents problem commands from running continuously and putting excess overhead on the agent computer. |

Filter Expression

Shell commands used in performance collection rules must return only a single numeric value, or errors will be encountered when collecting the performance value. The Filter Expression page allows you to filter the command output to ensure that the command output is acceptable. It is recommended that the default expression filter is used to only collect performance data when the value is numeric and the command executed successfully.

|  |  |
| --- | --- |
| Property Name | Description |
| Filter one or more events | An expression that filters output of the shell command. For more information on building expressions, see [Expressions](#z5e4020ce75c54c178719baeeafd546a5). The Parameter Name syntax for command execution output is: StdOut: //\*[local-name()="StdOut"] StdErr: //\*[local-name()="StdErr"] Return Code: //\*[local-name()="ReturnCode"] The default expression filters that the StdOut value is numeric, and that the script executed successfully, with the expression definition of: //\*[local-name()=”StdOut”] Matches Regular Expression ^[-+]?\d\*[0-9]\*(\.[0-9]+)?[Ee]?[-+]?[0-9]\*$ //\*[local-name()=”ReturnCode”] Equals 0 |

Performance Mapper

The Performance Mapper page defines the mapping of the command output to a performance counter.

|  |  |
| --- | --- |
| Option | Description |
| Object | Text for the Object name. This is required. You can type in the name of the object or select a property from the target. |
| Counter | Name of the performance counter. |
| Instance | Text for the Instance name. This only required if the performance counter has multiple instances. You can type in the name of the instance or select a property from the target. |
| Value | The variable that defines the value collected as a performance counter value. To collect the value returned by the command as StdOut, use $Data///\*[local-name()=”StdOut”]$. To collect the value returned by the command as ReturnCode, use $Data///\*[local-name()=”ReturnCode”]$ |

Creating UNIX/Linux Shell Command Performance Collection Rules

Use the following procedures to create a Windows performance collection rule in Operations Manager with the following details:

 Runs on all UNIX/Linux Computers, every 15 minutes

 Collects the count of files in the /tmp path as a performance counter

To create an UNIX/Linux shell command performance collection rule in Operations Manager

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then select Rules.  3. Right-click Rules and select Create a new rule.  4. On the Rule Type page, do the following:  a. Expand Collection Rules, expand Probe Based, and then click UNIX/Linux Shell Command (Performance).  b. Select the management pack from step 1.  c. Click Next.  5. On the General page, do the following:  a. In the Rule name box, type /tmp File Count.  b. In the Rule Category box, select Performance Collection.  c. Next to Rule Target, click Select and then select UNIX/Linux Computers.  d. Leave Rule is enabled selected.  e. Click Next.  6. On the Schedule page, do the following:  a. In the Run Every boxes, input 15 and Minutes.  b. Click Next.  7. On the Shell Command Details page, do the following:  a. In the Command box, type ls /tmp | wc –l. This command sequence will return the count of the files in /tmp.  b. In the Run As Profile box, select the UNIX/Linux Action Account profile.  c. In the Timeout (Seconds) box, input 120.  d. Click Next.  8. On the Filter Expression page, do the following:  a. Click Next (to use the default expression filter that validates StdOut is a numeric value, and the command executed successfully).  9. On the Performance Mapper page, do the following:  a. On the Object line, click […].  b. Click Target and select Network Name.  c. Click OK.  d. In the Counter box, type File Count.  e. In the Instance box, type /tmp.  f. In the Value box, type $Data///\*[local-name()=’StdOut’]$  g. Click Create. |

See Also

[Performance Monitors and Rules](#z7d91fa3c13794325a87bbef3aa7def9d)

[Performance Monitors](#z97cb90eebbaf4b5f8ca7094c89c5e7ba)

Script Monitors and Rules

Monitoring scripts are used when the required data cannot be collected through other standard means such as an event or performance counter. The script collects data from information on the agent and creates a property bag by using the MOM.ScriptAPI object that is installed with the Operations Manager agent.

Monitoring scripts may be written in any script language that can access the MOM.ScriptAPI object that is installed on all Operations Manager agents. You can use the Operations console to create scripts in VBScript or JScript. To use a Windows PowerShell script, you must use another Authoring tool such as the [Operations Manager R2 Authoring Console](#z3) or the [Visio Management Pack Designer](#z2).

Property Bags

Monitoring scripts send any output data as a property bag so that it can be evaluated in an expression for a monitor or mapped into performance data or an event for a collection rule. A property bag is a set of values that each has a name. Any name can be assigned although it is a best practice to use a name descriptive of the particular value. A property bag only exists during the life of the workflow. The next time that the workflow runs, the script is run and creates a new property bag with new values.

One property bag can have any number values, although the whole set of data may not exceed 4 MB. Most scripts will only require some values with a total size far under this limit. There is no requirement for all the values to be used by the workflow.

Scripts create property bags by using the CreatePropertyBag method on the MOM.ScriptAPI object. The workflow uses values from a property bag with a $Data variable that uses the following syntax:

$Data/Property[@Name="PropertyName"]

For example, a script creating performance data might create a property bag with values in the following table. This table shows the name of the value created by the script and the corresponding $Data variable that would be used to map the property bag data to performance data.

|  |  |  |
| --- | --- | --- |
| Property Bag Value Name | Sample Value | Variable |
| ObjectName | MyObject | $Data/Property[@Name='ObjectName']$ |
| CounterName | MyCounter | $Data/Property[@Name='CounterName']$ |
| InstanceName | MyInstance | $Data/Property[@Name='InstanceName']$ |
| Value | 10 | $Data/Property[@Name='Value']$ |

Script Structure

The following code shows a sample monitoring script to illustrate the basic structure of a monitoring script. This sample script has the following characteristics.

 Accepts arguments for the name of the computer that is running the script and a path of the location of the application.

 Creates a property bag with the values named ComputerName, InstanceName, and PerfValue.

sComputerName = WScript.Arguments(0)

sApplicationPath = WScript.Arguments(1)

Set oAPI = CreateObject("MOM.ScriptAPI")

Set oBag = oAPI.CreatePropertyBag()

oBag.AddValue "ComputerName", sComputerName

oBag.AddValue "InstanceName", "MyInstance"

oBag.AddValue "Value", 1.0

oAPI.Return(oBag)

Details of each section of the script are discussed here.

sComputerName = WScript.Arguments(0)

sApplicationPath = WScript.Arguments(1)

The first two lines of the script accept arguments. These values would be expected to be in the Arguments parameter of the rule or monitor running the script. The script can use any number of arguments that are required for the logic of the script.

Set oAPI = CreateObject("MOM.ScriptAPI")

Set oBag = oAPI.CreatePropertyBag()

The next two lines create a property bag. These lines will also be unchanged in most monitoring scripts. The main purpose of the rest of the script will be to add values to the property bag by using data that is collected from the agent computer.

oBag.AddValue "ComputerName", sComputerName

oBag.AddValue "InstanceName", "MyInstance"

oBag.AddValue "Value", 1.0

After the property bag is created, any number of values can be added to it. You do this with the AddValue method on the property bag object by using the name of the item followed its value. This example uses explicit values. In actual monitoring script, additional code would be expected that would collect information from the agent computer to include in these values.

oAPI.Return(oBag)

After all values are added to the property bag, it is returned into the workflow. This line is required, and without it the property bag is discarded when the script ends. This method is only used when the script creates only a single property bag. For more information about scripts that return multiple property bags and conditions when such a strategy is used, refer to the [Cookdown](http://go.microsoft.com/fwlink/?LinkID=232864) section of the [System Center Operations Manager 2007 R2 Authoring Guide](http://go.microsoft.com/fwlink/?LinkID=188119).

Script Arguments

Most scripts use arguments, which are values that are sent to the script from the command line when the script is run. Using arguments allows a single script to be used for multiple scenarios without modifying the script itself.

In a monitoring script, arguments are critical because there may be information that the script requires that will be different on each agent where the script runs. Any property of the target object for the monitor or rule can be used for the value of a script argument. This value is resolved individually on each agent at the time that the script is run.

Arguments are accessed in the Operations console from the Parameters button. Individual arguments should be separated by spaces in the order that they are accessed in the script. This is identical to the command line that would be provided if the script were run on a command line.

Each argument can be either an explicit value or a $Target variable to use the value of a property on the target object. Any $Target variables are resolved when the script is run so that the script is provided with the resolved values on the command line. You can type in the $Target variable if you know the proper syntax. It is easier though to select the property from Target button which will list all of the properties of the target object and its parents.

Important

Any $Target variable that might resolve to a value that includes a space should be enclosed with quotation marks. If a value includes spaces and does not have quotation marks, then it will be seen by the script as two separate arguments. The quotation marks will ensure that the value is seen as a single argument. If you select the property from Target menu, it will not include the quotation marks for you. You need to type these in after selecting the property.

For example, the sample script earlier expects two arguments for the computer name and the application path. Assuming this was part of a monitor or rule targeted at a class hosted by the Windows Computer class, the computer name could be retrieved from the PrincipalName property. If the application path were a property on the target class, then the arguments might look similar to the following example. Notice the quotation marks around the ApplicationPath property, because this could resolve to a value that contains a space.

$Target/Host/Property[Type="Windows!Microsoft.Windows.Computer"]/PrincipalName$ "$Target/Property[Type="MyApp.MyClass"]/ApplicationPath$"

Assuming that you gave the script a name of MyScript.vbs, the computer name was MyServer01, and the application path was C:\Program Files\Contoso\My Application, the command line that would be run for this script would be:

MyScript.vbs MyServer01 "C:\Program Files\Contoso\My Application"

Script Monitors and Rules Topics

 [Script Collection Rules](#z548dcbc638464d089268c55de8fbdf57)

Creating a rule that uses a script to collect performance or event data.

 [Script Monitors](#ze411a40fa71e47a99b41f5f1f8bfcfe4)

Creating a monitor that evaluates the results of a script to set its health state.

 [UNIX/Linux Shell Command Monitors](#z702606572e1b4e89b501cf7a8c3b2d5d)

Creating a monitor that evaluates the output of execution of an UNIX/Linux command, script, or one-line sequence of multiple commands (using pipeline operators).

Script Collection Rules

Script collection rules run a script on a schedule and store its output as either performance data or an event. As part of creating the rule, you need to specify which property bag values from the script or properties from the target object that will be used for different properties of the event or performance data being created.

Options

When you run the script collection wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

|  |  |
| --- | --- |
| Option | Description |
| Rule Name | The name used for the rule. This appears in the Rules view in the Authoring pane. |
| Description | Optional description of the rule. |
| Management Pack | Management pack to store the monitor.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Rule Category | The category for the rule. For an event collection rule, this should be Event Collection. For a performance collection rule, this should be Performance Collection. |
| Rule target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |

Schedule

The Schedule page defines the schedule to run the script. The script will run indefinitely according to this schedule until the rule is disabled or deleted or the management pack is uninstalled.

|  |  |
| --- | --- |
| Option | Description |
| Run every | Frequency that the script should be run. This should typically not be less than 5 minutes. |
| Synchronize at | If enabled, the schedule will be synchronized to occur at the specified time. |

Script

The Script page contains the body of the script itself and its parameters. You can type the script directly into the dialog box, but you will usually write it using another text editor and then copy the text of the script and paste it. This allows you to use a more functional editing tool and test the script on a command line before including it in the management pack.

|  |  |
| --- | --- |
| Option | Description |
| File Name | Name of the script. Must have either a .vbs or .js extension depending on its language. There is no requirement to make this name unique because each script is provided its own temporary directory on the agent. |
| Timeout | The number of seconds that the script can run before the agent stops it. This prevents problem scripts from running continuously and putting excess overhead on the agent computer.  The timeout value assigned to a script should allow enough time for the script to run under ordinary conditions, but should be less than the interval that the script is scheduled to run. If a script is configured to have a timeout value greater than its duration, then possibly multiple copies of the script could be running concurrently. |
| Script | The body of the script. |
| Parameters | Click to provide values for any arguments in the script. For more information, see [Script Arguments](#z78). |

Performance Mapper (Performance Collection Only)

The Performance Mapper page is used to define values for the properties of the performance data being collected.

|  |  |
| --- | --- |
| Option | Description |
| Object | Text for the Object name. This is required. |
| Counter | Text for the Counter name. This is required. |
| Instance | Text for the Instance name. This only required if the target of the rule has multiple instances. |
| Value | Numeric for the value for the performance |

Event Mapper (Event Collection Only)

The Event Mapper page is used to define values for the properties of the event that will be collected. The value for each field will either be an explicit string of text, a value from the property bag of the script, or the value of a property of the target object.

|  |  |
| --- | --- |
| Option | Description |
| Computer | The name of the computer that the event was logged on. This will usually be a $Target variable for the Principal Name of the computer. You can select this value by clicking on the ellipse button next to the text box. |
| Event source | The source of the event. This will usually be an explicit value but may be a $Data variable to use the value of a property from the script. |
| Event log | The name of the event log. This will usually may be an explicit value or a $Data variable to use the value of a property from the script. |
| Event ID | The numeric event number. This will usually be an explicit value or a $Data variable to use the value of a property from the script. |
| Category | The value of the EventCategory parameter (an integer from 0 to 65535) is an index into a category dynamic-link library (DLL) message table that contains a localized string. Each publisher defines its own set of categories. These categories commonly correspond to individual components (for example: a connector, module host, or data warehouse). |
| Level | The severity of the event. You can select this value from the drop down list. |

Creating Script Collection Rules

The following procedure creates a performance script collection rule with the following details:

 Runs on any computer with an instance of a particular service installed.

 The script accepts two parameters, one for computer name and another for the version of the application that is stored as a property on the target class.

 The script itself is only for testing and performs no real function. It simulates a script running a synthetic transaction and returning a property bag with static values.

To create a script based performance collection rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Rules.  4. Right-click Rules and select Create a new rule.  5. On the Rule Type page, do the following:  a. Expand Collection Rules, then expand Probe Based, and then select Script (Performance).  b. In the Management Pack dropdown, select the management pack from step 1.  c. Click Next  6. On the General page, do the following:  a. In the Rule name box, type My Application Collect Script Performance.  b. In the Rule Category drop down box, select Performance Collection.  c. Click Select.  d. Select the name of the target you created in step 2.  e. Click OK.  7. On the Schedule page, do the following:  a. In the Run every box, type 15 minutes.  b. Click Next  8. On the Script page, do the following:  a. For the File Name value, type MyPerfCollectionScript.vbs  b. For the Timeout value, type 1 minutes  c. In the Script box, paste the complete contents of the following script.  sComputerName = WScript.Arguments(0)  sVersion = WScript.Arguments(1)  [oAPI.LogScriptEvent]  Set oAPI = CreateObject("MOM.ScriptAPI")  Set oBag = oAPI.CreatePropertyBag()  Call oBag.AddValue("ComputerName",sComputerName)  Call oBag.AddValue("InstanceName","MyInstance")  Call oBag.AddValue("Value",10)  oAPI.Return(oBag)  d. Click the Parameters button.  e. Select Target, select (Host=Windows Computer), and then select Principal Name (Windows Computer).  f. Type a SPACE.  g. Select Target and then Version (My Computer Role Base).  h. Click OK.  i. Click Next.  9. On the Performance Mapper page, do the following:  a. In the Object box type MyApplication.  b. In the Counter box type MyCounter.  c. In the Instance box type $Data/Property[@Name=FileName]$.  d. In the Value box type $Data/Property[@Name=’FileSize’]$.  e. Click Create. |

The following procedure creates an event script collection rule with the following details:

 Runs on any computer with an instance of a particular service installed.

 The script accepts two parameters, one for computer name and another for the version of the application that is stored as a property on the target class.

 The script itself is only for testing and performs no real function. It simulates a script running a synthetic transaction and returning a property bag with static values.

To create a script based event collection rule

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Rules.  4. Right-click Rules and select Create a new rule.  5. On the Rule Type page, do the following:  a. Expand Collection Rules, then expand Probe Based, and then select Script (Event).  b. In the Management Pack dropdown, select the management pack from step 1.  c. Click Next  6. On the General page, do the following:  a. In the Rule name box, type My Application Collect Script Event.  b. In the Rule Category drop down box, select Event Collection.  c. Click Select.  d. Select the name of the target you created in step 2.  e. Click OK.  7. On the Schedule page, do the following:  a. In the Run every box, type 15 minutes.  b. Click Next  8. On the Script page, do the following:  a. For the File Name value, type MyEventCollectionScript.vbs  b. For the Timeout value, type 1 minutes  c. In the Script box, paste the complete contents of the following script.  sComputerName = WScript.Arguments(0)  sVersion = WScript.Arguments(1)  Set oAPI = CreateObject("MOM.ScriptAPI")  Set oBag = oAPI.CreatePropertyBag()  Call oBag.AddValue("ComputerName",sComputerName)  Call oBag.AddValue("EventID",100)  Call oBag.AddValue("ParamValue","Param1")  oAPI.Return(oBag)  d. Click Parameters.  e. Select Target, select (Host=Windows Computer), and then select Principal Name (Windows Computer).  f. Type a SPACE.  g. Select Target and then Version (My Computer Role Base).  h. Click OK.  i. Click Next.  9. On the Event Mapper page, do the following:  a. In the Computer box type $Data/Property[@Name='ComputerName']$.  b. In the Event source box type MyApp.  c. In the Event log box type CustomScript.  d. In the Event ID box type $Data/Property[@Name='EventID']$.  e. In the Category box type 0.  f. In the Level box select Information.  g. Click the Parameters button.  h. Type $Data/Property[@Name='ParamValue']$  i. Click OK.  j. Click Create. |

See Also

[Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28)

Script Monitors

Script monitors run a script on a schedule and use its output to determine the health state of the target object. Script monitors are useful for performing test transactions against applications or gathering information that is not accessible through other means. The results of the script are returned in a [Property Bag](#z76) that are evaluated against criteria to determine the resulting health state.

[Conceptual view of script monitor]

Options

When you run a script monitor wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

General

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the monitor. This appears in the Health Explorer for each target object. |
| Description | Optional description of the monitor. |
| Management Pack | Management pack to store the classes, monitors, and rules created by the template.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Monitor target | The class to use for the target of the monitor. The monitor will be run on any agent that has at least one instance of this class, and the health of those objects will be affected by the health of this monitor. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Parent Monitor | The aggregate monitor that this monitor will be placed under in the Health Explorer. |
| Monitor is enabled | If checked, the monitor is enabled and the script will run according to the schedule.  If unchecked, the monitor is not enabled and the script will not run. The monitor can be enabled for a group of target objects by creating an override to enable the monitor. |

Schedule

The Schedule page defines the schedule to run the script. The script will run indefinitely according to this schedule until the monitor is disabled or deleted or the management pack is uninstalled.

|  |  |
| --- | --- |
| Option | Description |
| Run every | Frequency that the script should be run. This should typically not be less than 5 minutes. |
| Synchronize at | If enabled, the schedule will be synchronized to occur at the specified time. |

Script

The Script page contains the body of the script itself and its parameters. You can type the script directly into the dialog box, but you will usually write it using another text editor and then copy the text of the script and paste it. This allows you to use a more functional editing tool and test the script on a command line before including it in the management pack.

|  |  |
| --- | --- |
| Option | Description |
| File Name | Name of the script. Must have either a .vbs or .js extension depending on its language. There is no requirement to make this name unique because each script is provided its own temporary directory on the agent. |
| Timeout | The number of seconds that the script can run before the agent stops it. This prevents problem scripts from running continuously and putting excess overhead on the agent computer.  The timeout value assigned to a script should allow enough time for the script to run under ordinary conditions, but should be less than the interval that the script is scheduled to run. If a script is configured to have a timeout value greater than its duration, then possibly multiple copies of the script could be running concurrently. |
| Script | The body of the script. |
| Parameters | Click to provide values for any arguments in the script. For more information, see [Script Arguments](#z78). |

Expressions

Each required expression for the monitor will have its own page in the wizard. A two state monitor will have the following expressions:

 Unhealthy Expression

 Healthy Expression

A three state monitor will have the following expressions:

 Unhealthy Expression

 Degraded Expression

 Healthy Expression

Note

Detailed information on expressions is available in [Expressions](#z5e4020ce75c54c178719baeeafd546a5).

Each expression will typically compare the value of one or more of the properties from the script’s property bag to some value. Each expression must be different, and only one of the expressions should evaluate to True under any particular condition. In the next page of the wizard, you will associate each of the health states of the monitor with one of these expressions. When an expression evaluates to the True, the monitor will be set to that health state.

For example, the script might perform a test transaction against a particular application and return a single property with a value of “Good” if the transaction completed successfully, and “Bad” if the transaction failed.

|  |  |
| --- | --- |
| Option | Description |
| Parameter Name | This will be a $Data variable representing the particular value that you need from the property bag. This will be in the following syntax:  Property[@Name="PropertyName"] |
| Operator | The type of comparison to perform. |
| Value | The explicit value that should match the value in the property bag. |

Configure Health

On this page, you map each of the expressions to a health state for the monitor. When a condition is true, the monitor is set to the health state that you define. For a three state monitor, you can typically accept the default settings. For a two state monitor, you typically only have to determine if the Unhealthy Expression should result in a Critical or Warning state.

|  |  |
| --- | --- |
| Option | Description |
| Monitor Condition | Represents each of the expressions. |
| Health State | The health state to set the monitor to when that expression is true. |

Configure Alerts

|  |  |
| --- | --- |
| Option | Description |
| Generate alerts for this monitor | If checked, an alert will be created when the monitor changes from a healthy state to a warning or critical state, and all of the other options will be enabled.  If unchecked, the monitor will not generate alerts when the health state is changed, and all of the other options will be disabled. |
| Generate an alert when | For a two-state monitor, this setting should be set to The monitor is in a critical health state. |
| Automatically resolve alert when the monitor returns to a healthy state | If checked, the alert will automatically be resolved when the monitor returns to a healthy state. If unchecked, the alert must be resolved manually. |
| Alert name | The name of the alert that is displayed in the console. |
| Alert Description | The description of the alert. |
| Priority | The priority of the alert: Low, Medium, or High. |
| Severity | The severity of the alert: Information, Warning, Critical, or matched to the health state of the monitor. |

Creating a Script Monitor

The following procedure shows how to create a monitor based on a monitoring script with the following details:

The monitor created in this procedure has the following characteristics:

 Runs on any computer with an instance of a particular service installed.

 Sets the monitor to a critical state when the script returns a status message of Bad.

 Sets the monitor to a healthy state when the script returns a status message of Good.

 The script accepts an argument for the computer name of the target object’s agent and for an argument specifying wherther thereturns a Good or Bad message.

 The script itself is only for testing and performs no real function. It simulates a script running a synthetic transaction.

To create a two state script monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. Create a new target using the process in [To create a Windows Service template](#z63). You can use any service installed on a test agent for this template.  3. In the Operations console, select the Authoring workspace, and then select Monitors.  4. In the Monitors pane, click Change Scope and then select the name of the target that you created in step 2.  5. Expand the target class then expand Entity Health.  6. Right-click Availability, select Create a Monitor, and then select Unit Monitor.  7. On the Monitor Type page, do the following:  a. Expand Scripting and then expand Generic.  b. Select Timed Script Two State Monitor.  c. In the Management Pack dropdown, select the management pack from step 1.  d. Click Next.  8. On the General page, do the following:  a. In the Name box, type My Application Script Monitor.  b. The Monitor target box should already have the correct target class.  c. Parent Monitor box should already have Availability.  d. Leave the Monitor is enabled box selected.  e. Click Next.  9. On the Schedule page, do the following:  a. In the Run every box, type 15 minutes.  b. Click Next  10. On the Script page, do the following:  a. For the File Name value, type MyScript.vbs  b. For the Timeout value, type 1 minutes  c. In the Script box, paste the complete contents of the following script.  sComputerName = WScript.Arguments(0)  bTestSuccessful = WScript.Arguments(1)  Set oAPI = CreateObject("MOM.ScriptAPI")  oAPI.LogScriptEvent "MyScript.vbs",10,4, "Running script on " & sComputerName  Set oBag = oAPI.CreatePropertyBag()  Call oBag.AddValue("ComputerName",sComputerName)  If bTestSuccessful = True Then  Call oBag.AddValue("Result","Good")  Else  Call oBag.AddValue("Result","Bad")  End If  oAPI.Return(oBag)  d. Click the Parameters button.  e. Select Target, then select (Host=Windows Computer), then select Principal Name (Windows Computer).  f. Type a space after the Principal Name variable and then type False.  g. Click OK.  h. Click Next.  11. On the Unhealthy Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type Property[@Name='Result'].  c. In the Operator box select Equals.  d. In the Value box type Bad.  e. Click Next.  12. On the Healthy Expression page, do the following:  a. Click Insert.  b. In the Parameter Name box type Property[@Name='Result'].  c. In the Operator box select Equals.  d. In the Value box type Good.  e. Click Next.  13. On the Configure Health page, do the following:  a. Change the Health State for the Unhealthy condition to Critical.  b. Click Next.  14. On the Configure Alerts page, do the following:  a. Select Generate alerts for this monitor  b. In the Generate an alert when box, select The monitor is in a critical health state.  c. Leave the box selected to automatically resolve the alert.  d. In the Alert name box, type Application test failed.  e. Clear the existing text in the Alert description box and type Result:  f. Click the ellipse button.  g. Click Data and then Property.  h. Replace <<STRING>> with Result.  i. Click OK.  15. Click Create. |

See Also

[Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28)

UNIX/Linux Shell Command Monitors

UNIX/Linux shell command monitors run on a schedule and execute a program or script, a command, or a one-line command sequence (using pipeline operators). The output from the command is used to determine the health state of the target object. Shell command monitors are useful for custom monitoring of UNIX and Linux applications with information that is not accessible through other means.

Options

When you run an UNIX/Linux shell command monitor wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

Rule Type

The Rule Type page includes basic settings for the rule including its type and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Select the type of monitor to create | To create a shell command monitor that evaluates for two states (healthy and error), select UNIX/Linux Shell Command Two State Monitor. To create a shell command monitor that evaluates for three states (healthy, warning and error), select UNIX/Linux Shell Command Three State Monitor. |
| Management Pack | Management pack file to store the rule or monitor.For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

General

The General page includes general settings for the rule including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Name | The name used for the monitor. This appears in the Monitors view in the Authoring pane. When you view the Health Explorer for the monitoring target, you can see the health state of this monitor. |
| Description | Optional description of the rule. |
| Monitor Target | The class to use for the target of the rule. The rule will be run on any agent that has at least one instance of this class. For more information on targets, see [Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f). |
| Parent Monitor | The aggregate monitor that this monitor will be placed under in the Health Explorer. |
| Monitor is enabled | If checked, the monitor is enabled and the shell command will run according to the schedule. If unchecked, the monitor is not enabled and the script will not run. The monitor can be enabled for a group of target objects by creating an override to enable the monitor. |

Schedule

The Schedule page defines the schedule to run the script. The script will run indefinitely according to this schedule until the monitor is disabled or deleted or the management pack is uninstalled.

|  |  |
| --- | --- |
| Option | Description |
| Run every | Frequency that the script should be run. This should typically not be less than 5 minutes. |
| Synchronize at | If enabled, the schedule will be synchronized to occur at the specified time. |

UNIX/Linux Shell Command

The following options are available on the Shell Command Details page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Command | The shell command to execute. This can be the full path to a program or script, a command, or a one-line sequence of multiple commands (using pipeline operators). |
| Run As Profile | Either the “UNIX/Linux Action Account” or “UNIX/Linux Privileged Account” profile. Select the profile that associates the required account credentials with the task target. The associated account will be used to execute the command. |
| Timeout (seconds) | The number of seconds that the command can run before the agent stops it. This prevents problem commands from running continuously and putting excess overhead on the agent computer. |

Expressions

Each required expression for the monitor will have its own page in the wizard. A two-state monitor will have the following expressions:

1. Error Expression

2. Healthy Expression

A three-state monitor will have the following expressions:

1. Error Expression

2. Warning Expression

3. Healthy Expression

Note

Detailed information on expressions is available in [Expressions](#z5e4020ce75c54c178719baeeafd546a5).

Each expression will typically compare the value of one or more of the properties from the command’s output to some value. Each expression must be different, and only one of the expressions should evaluate to True under any particular condition. In the next page of the wizard, you will associate each of the health states of the monitor with one of these expressions. When an expression evaluates to the True, the monitor will be set to that health state.

|  |  |
| --- | --- |
| Option | Description |
| Parameter Name | The Parameter Name syntax for command execution output is:StdOut: //\*[local-name()="StdOut"]StdErr: //\*[local-name()="StdErr"]Return Code: //\*[local-name()="ReturnCode"] |
| Operator | The type of comparison to perform. |
| Value | The explicit value that should match the value in the property bag. |

Configure Health

On this page, you map each of the expressions to a health state for the monitor. When a condition is true, the monitor is set to the health state that you define. For a three-state monitor, you can typically accept the default settings. For a two-state monitor, you typically only have to determine if the Unhealthy Expression should result in a Critical or Warning state.

|  |  |
| --- | --- |
| Option | Description |
| Monitor Condition | Represents each of the expressions. |
| Health State | The health state to set the monitor to when that expression is true. |

Configure Alerts

|  |  |
| --- | --- |
| Option | Description |
| Generate alerts for this monitor | If checked, an alert will be created when the monitor changes from a healthy state to a warning or critical state, and all of the other options will be enabled. If unchecked, the monitor will not generate alerts when the health state is changed, and all of the other options will be disabled. |
| Generate an alert when | For a two-state monitor, this setting should be set to The monitor is in a critical health state. |
| Automatically resolve alert when the monitor returns to a healthy state | If checked, the alert will automatically be resolved when the monitor returns to a healthy state. If unchecked, the alert must be resolved manually. |
| Alert name | The name of the alert that is displayed in the console. |
| Alert Description | The description of the alert. |
| Priority | The priority of the alert: Low, Medium, or High. |
| Severity | The severity of the alert: Information, Warning, Critical, or matched to the health state of the monitor. |

Creating an UNIX/Linux Shell Command Monitor

The following procedure shows how to create a monitor based on a monitoring shell command with the following details:

 Runs on any UNIX/Linux computer every 15 minutes

 Sets the monitor to a Critical state when the file “/tmp/error” exists.

 Sets the monitor to a Healthy state when the file “/tmp/error" does not exist.

To create a two state UNIX/Linux shell command monitor

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace, and then select Monitors.  3. Launch the Create a Monitor task, and select Unit Monitor.  4. On the Monitor Type page, do the following:  a. Expand Scripting and then expand Generic.  b. Select UNIX/Linux Shell Command Two State Monitor.  c. In the Management Pack dropdown, select the management pack from step 1.  d. Click Next.  5. On the General page, do the following:  a. In the Name box, type Error File Test Monitor.  b. Click Select next to the Monitor Target box, select UNIX/Linux Computer, and click OK.  c. Select the Parent Monitor of Availability.  d. Leave the Monitor is enabled box selected.  e. Click Next.  6. On the Schedule page, do the following:  a. In the Run every box, type 15 minutes.  b. Click Next.  7. On the Shell Command Details page, do the following:  a. In the Command box, type ls /tmp/error | wc –l. This command sequence will return a 1 if the file “/tmp/error” exists, and a 0 if it does not.  b. In the Run As Profile box, select the UNIX/Linux Action Account profile.  c. In the Timeout (Seconds) box, input 120.  d. Click Next.  8. On the Error Expression page, enter the following And expression:  a. //\*[local-name()=”StdOut”] equals 1  b. //\*[local-name()=”ReturnCode”] equals 0  9. On the HealthyExpression page, enter the following And expression:  a. //\*[local-name()=”StdOut”] does not equal 1  b. //\*[local-name()=”ReturnCode”] equals 0  10. On the Configure Health page, do the following:  a. Leave the Health State for the StatusError condition set to Critical.  b. Click Next.  11. On the Configure Alerts page, do the following:  a. Select Generate alerts for this monitor.  b. In the Generate an alert when box, select The monitor is in a critical health state.  c. Leave the box selected to automatically resolve the alert.  d. In the Alert name box, type Application test failed.  e. Clear the existing text in the Alert description box and type Computer:  f. Click the ellipse button.  g. Click Target and then select Network Name.  h. Click OK.  i. In the Alert description box, add a new line and type StdOut: $Data/Context///\*[local-name()=”StdOut”]$  12. Click Create. |

See Also

[Script Monitors and Rules](#z26b778f21255487c868ab09c7fc5ab28)

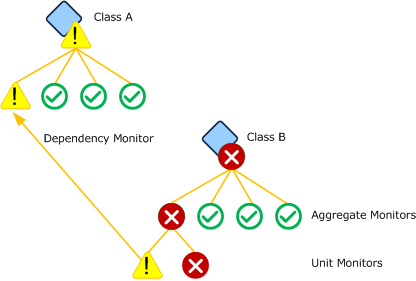
Dependency Monitors

Dependency monitors let the health of one object be affected by the health of another object. This allows for health rollup between specific related instances of different classes.

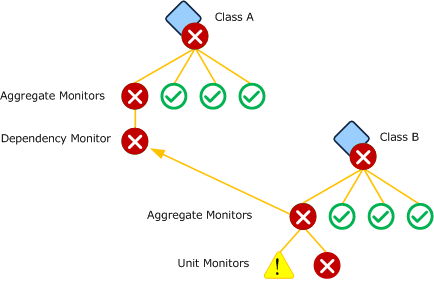
Each dependency monitor is based on a specific hosting or containment relationship. Just creating a relationship between two objects does not alone provide rollup between their health states. A dependency monitor must be associated with the relationship for rollup of health to be performed.

The source and target class for a dependency monitor are defined by the relationship that the monitor is based on. The monitor must additionally specify a specific unit monitor or aggregate monitor on the target class and an aggregate monitor on the source class. Only the health of the target monitor is considered when calculating the health of the dependency monitor, and it only affects the health of the specified aggregate monitor on the target object.

Dependency monitor based on unit monitor

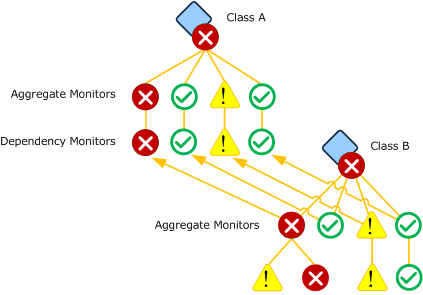


Dependency monitor based on aggregate monitor



Multiple dependency monitors can be created on a single relationship if the health of the source class should be affected by multiple unit or aggregate monitors on the target class. For example, a dependency monitor might be created for each standard aggregate monitor as shown in the following image.

Multiple dependency monitors for a single class



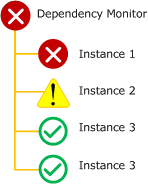
Health Rollup Policy

There may be multiple instances of the target class, each with a different health state. Each dependency monitor must define a health rollup policy to define the logic that is used to determine the health of the dependency monitor based on the health of the instances of its target monitor. The possible health rollup policies for a dependency monitor are as follows:

Worst state policy

The source object matches the state of the target object that has the worst health state. This is used when the source object should only be healthy if all the target objects are healthy. This is the most common policy used by dependency monitors.

Worst state health policy

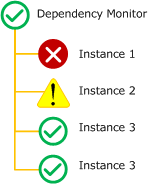


Best state policy

The source object matches the state of the target object that has the best health state. This policy is used when only one of the source objects has to be healthy for the target object to be healthy.

For example, the Microsoft Windows Hyper-V 2008 Monitoring management pack has a dependency monitor on the hosting relationship from Microsoft.Windows.HyperV.ServerRole to Microsoft.Windows.HyperV.VirtualNetwork that uses a best state policy. This is because the server running Hyper-V is functional as long as it has one functional virtual network. The logic defined by this management pack is that the server class should show an error state if no virtual networks are available.

Best state health policy

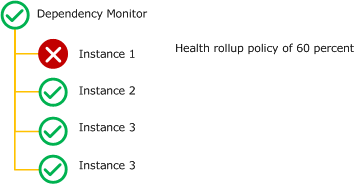


Percentage policy

The source object matches the worst state of a single member of a specified percentage of target objects in the best state. This policy is used when a certain percentage of target objects must be healthy for the target object to be considered healthy.

For example an application might run on a web farm that includes multiple Web servers. Because of the redundancy offered in this kind of deployment, the application might be considered healthy if a particular percentage of servers is available. The farm itself could be represented in the management pack by a health rollup class based on System.ApplicationComponent with a containment relationship to the Web servers. A dependency monitor could be created on this containment relationship with a health rollup policy specifying a percentage. Even if one or more Web servers had a problem, as long as the specified percentage were in a healthy state then the class representing the web farm would also be healthy.

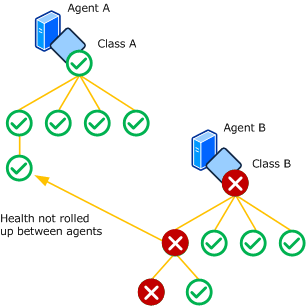
Percentage health policy



Health rollup between agents

Health state can only be rolled up between objects managed by the same agent unless the source object is managed by the Root Management Server. Groups and classes used for health rollup are typically unhosted. This means that they are managed by the RMS so that they can roll up health from objects managed by different agents. A relationship can be discovered between objects managed by different agents, but any dependency monitor associated with that relationship will not work as expected.

Health rollup between agents



Aggregate Monitors

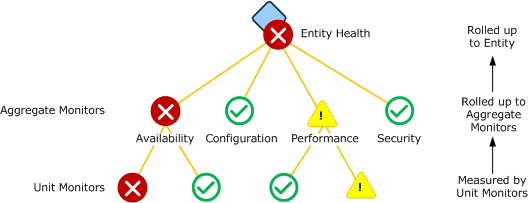
Aggregate monitors group multiple monitors to provide a single health aggregated health state. This provides an organization to all of the monitors targeted at a particular class and provides a consolidated health state for specific categories of operation.

Standard Aggregate Monitors

Every class has four standard aggregate monitors: Availability, Configuration, Performance, and Security. These are in the System.Health.Library management pack and targeted at the Entity class. Because all classes inherit from the Entity class, all classes inherit these standard monitors. The standard set of aggregate monitors will be sufficient for most classes.

Most monitors will fall into one of the four categories represented by the standard aggregate monitors. Because of this, custom aggregate monitors will typically use one of the standard aggregate monitors as their parent instead of being positioned alongside them directly under the entity health. Unit monitors and dependency monitors will similarly use either a custom aggregate monitor or one of the standard aggregate monitors as their parent.

Standard aggregate monitors



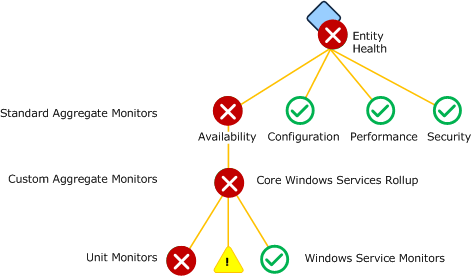
Custom Aggregate Monitors

Management packs can include custom aggregate monitors specific to the requirements of classes in a particular application. These monitors may use another aggregate monitor for their parent or the top level Entity State similar to the standard aggregate monitors use. Custom aggregate monitors can be configured underneath another aggregate monitor or attached directly to the entity state.

For example, the Windows Server 2008 Operating System (Monitoring) management pack includes an aggregate monitor called Microsoft.Windows.Server.2008.OperatingSystem.CoreServicesRollup that is used to combine the health of the different services that are monitored by this management pack. There are nine services that the management pack considers critical to the operation of a computer running Windows Server 2008. Instead of positioning these directly under the Availability aggregate monitor alongside other unit monitors, the aggregate monitor provides a combined health measurement for all the related services.

This aggregate monitor is illustrated in the following diagram.

Core Windows Services Rollup aggregate monitor



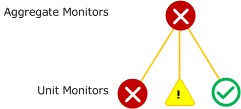
Health Rollup Policy

Each aggregate monitor must define a health rollup policy which is the logic that is used to determine the health of the aggregate monitor based on the health of the monitors under it. The possible health rollup policies for an aggregate monitor are as follows:

Worst state

The state of the aggregate monitor matches the state of the child monitor with the worst health state. This is the most common policy used by aggregate monitors.

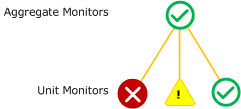
Worst state health policy



Best state

The state of the aggregate monitor matches the state of the child monitor with the best health state.

Best state health policy



Alerting on Aggregate Monitors

You can configure alerts on aggregate monitors just like with unit monitors, and you can leverage this feature to reduce noise by limiting the number of redundant alerts you receive. To implement this strategy, group together a common set of unit monitors underneath an aggregate monitor. Configure the aggregate monitor to send an alert, but configure unit monitors to not send an alert. When one of the unit monitors goes to a critical or warning state (depending on the configuration of your aggregate monitor), it will cause the aggregate monitor to change to the same state and cause an alert to be sent. If another unit monitor under the same aggregate goes to a warning or critical state, another alert won’t be sent since the aggregate monitor will already be in the negative state.

This strategy should only be used if the unit monitors measure a particular feature of the application. The assumption it relies on is that a problem with one of the monitors will alert support staff to a problem in the application. They can inspect the Health Explorer for further information including the other monitors that may be unhealthy. Additional alerts for a similar problem would not be of value but only create excess noise.

If the unit monitors measure different features of the application such that alerts should be received separately from each, then either multiple aggregate monitors should be created or this strategy should not be used.

Diagnostics and Recoveries

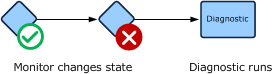
Diagnostics and recoveries run when a monitor changes state. Diagnostics collect additional information about the detected problem. Recoveries try to resolve the problem. Each will typically run a command or script that outputs information displayed in the Health Explorer in the Operations Console.

Diagnostics

Diagnostics run after a monitor changes state and try to collect additional information about the issue. This information is provided to the user with the state change history in the properties of the monitor. If the Run diagnostic automatically property of the diagnostic is set to true, then it is run automatically when the monitor changes state. If the Run diagnostic automatically property of the diagnostic is set to false, then a link is provided to the user in the Operations console that they can click to run the diagnostic.

Diagnostics are not intended to make any changes to the application or system that they are running on. Because they are running a script or command, however, there is no way for Operations Manager to make sure that these changes are not being made. It is the responsibility of the management pack author to make sure that no such changes are being made. If changes to the application or system are required, then a recovery should be used.

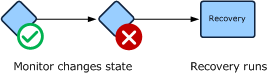
Diagnostic



Recoveries

Recoveries are workflows that run after a monitor changes state. Recoveries try to correct the issue, and return the monitor to a healthy state. Any output from the recovery is provided to the user with the state change history in the properties of the monitor that the diagnostic is associated with. If the Run diagnostic automatically property of the recovery is set to true, then the recovery is run automatically when the monitor changes state. If the Run diagnostic automatically property of the recovery is set to false, then a link is provided to the user in the Operations console that they can click to run the recovery.

Recovery



Recalculating State

If a recovery is successful, then the monitor should return to a healthy state the next time that the monitor detects the required information. If the monitor runs a scheduled script for example, then the monitor will return to healthy the next time that the script runs and the monitors detects the criteria for a healthy state. If the monitor relies on an event for its healthy state, the application is expected to create the required event in response to the recovery successfully correcting the problem. If a monitor is configured to use a manual reset though, then the user will still be required to manually set it to a healthy state.

A recovery can be configured to recalculate the state of the monitor immediately after it runs. This option has the same effect as the user selecting Recalculate Health for the monitor in the Operations console. Recalculating state only has an effect on monitors that run on a schedule such as a script and that have on demand detection defined. If the monitor does not have on demand detection defined, then the option has no effect. The advantage of configuring a monitor to recalculate state is that it can return the monitor to a healthy state immediately instead of waiting for the schedule.

Diagnostic and Recovery Wizard Options

When you run the Diagnostic or Recovery wizard, you will need to provide values for options in the following tables. Each table represents a single page in the wizard.

Diagnostic or Recovery Task Type

The Diagnostic Task Type or Recovery Task Type page allows you to specify the type of diagnostic or recovery to create and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Select the type of diagnostic task to create  Select the type of recovery task to create | Specifies if the diagnostic or recovery will run a command or a script. |
| Management Pack | Management pack file to store the diagnostic or recovery. If the monitor is in an unsealed management pack file, then the diagnostic or recovery must be stored in the same one.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |

General

The General page includes general settings for the diagnostic or wizard including its name, category, target, and the management pack file to store it in.

|  |  |
| --- | --- |
| Option | Description |
| Diagnostic Name  Recovery Name | The name used for the diagnostic or recovery. For a rule, the name appears in the Rules view in the Authoring pane. When you create a view or report, you can select this name to use the data collected by it. For a monitor, the name appears in the Health Explorer of any target objects. |
| Description | Optional description of the diagnostic or recovery. |
| Management Pack | Management pack to store the rule.  For more information on management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c). |
| Select the health state for which this diagnostic will run  Select the health state for which this recovery will run | Specifies the health state that the diagnostic or recovery will run in response to. When the monitor changes to the specified state from another state, the diagnostic or recovery will run. |
| Run diagnostic automatically  Run recovery automatically | If selected, the diagnostic or recovery will automatically run when the monitor changes to the specified state. If not selected, the diagnostic or recovery will be displayed as a link in the Health Explorer that the user can click to run. |
| Recalculate monitor state after recovery finishes  (Recoveries only) | Specifies whether the recovery should initiate a recalculation of the monitor health state after completing. This is described in [Recalculating State](#z79). |

Command Line

The Command Line page is only available for Run Command diagnostics and recoveries. It includes details about the command to run.

|  |  |
| --- | --- |
| Option | Description |
| Full path to file | The path and name of the executable to run. |
| Parameters | Any command line parameters that you would include after the executable if you were running it from a command line. You can include variables that refer to values from properties of the target object of the monitor. |
| Working directory | The default directory to set while the command runs if the executable requires one. |
| Timeout (in seconds) | The number of seconds that the command can run before the agent stops it. This prevents problem scripts from running continuously and putting excess overhead on the agent computer.  The timeout value assigned to a script should allow enough time for the script to run under ordinary conditions, but should be less than the interval that the script is scheduled to run. If a script is configured to have a timeout value greater than its duration, then possibly multiple copies of the script could be running concurrently. |

Script

The Script page is only available for Run Script diagnostics and recoveries. It contains the body of the script itself and its parameters. You can type the script directly into the dialog box, but you will usually write it using another text editor and then copy the text of the script and paste it. This allows you to use a more functional editing tool and test the script on a command line before including it in the management pack.

|  |  |
| --- | --- |
| Option | Description |
| File Name | Name of the script. Must have either a .vbs or .js extension depending on its language. There is no requirement to make this name unique because each script is provided its own temporary directory on the agent. |
| Timeout | The number of seconds that the script can run before the agent stops it. This prevents problem scripts from running continuously and putting excess overhead on the agent computer.  The timeout value assigned to a script should allow enough time for the script to run under ordinary conditions, but should be less than the interval that the script is scheduled to run. If a script is configured to have a timeout value greater than its duration, then possibly multiple copies of the script could be running concurrently. |
| Script | The body of the script. |
| Parameters | Click to provide values for any arguments in the script. For more information, see [Script Arguments](#z78). |

Creating Diagnostics and Recoveries

Use the following procedure to create a diagnostic in Operations Manager with the following details:

 Collects a list of the processes running on the agent computer.

To create a diagnostic that runs a command

|  |
| --- |
| 1. Create a monitor using a procedure from one of the topics in [Monitors and Rules](#zdc82139bfb7f47b5ae3fcafeecf4d4ff).  In the Operations console, select the Authoring workspace, and then select Monitors.  2. In the Monitors pane, expand the monitor you created in the first step.  3. Right-click the monitor and select Properties.  4. Select the Diagnostic and Recovery tab.  5. Click Add and then Diagnostic for a warning health state or Diagnostic for a critical health state. The options available in this step will depend on the health states of the monitor.  6. On the Diagnostic Task Type page, do the following:  a. In the Select the type of diagnostic task to create box, select Run Command.  b. Click Next.  7. On the General tab, do the following:  a. In the Diagnostic Name box, type Check Running Processes.  b. In the Select the health state for which this diagnostic will run dropdown, select either Warning or Critical. The diagnostic will run when the monitor changes from a healthy state to the specified state.  c. Ensure that Run diagnostic automatically is selected.  d. Click Next.  8. On the Command Line tab, do the following:  a. In the Full path to file box, type %windir%\system32\tasklist.exe.  b. Click Create.  c. Click OK. |

Tasks

Tasks in System Center 2012 – Operations Manager are actions that can be run on demand by the user. Depending on the kind of task, the action may run either on the user’s local workstation or on one or more specified agents.

In This Section

Console Tasks

|  |
| --- |
| Tasks that run on the user’s workstation using the current user’s credentials. |

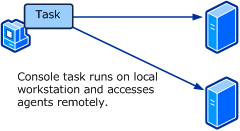
Agent Tasks

|  |
| --- |
| Tasks that run on the agent computer using the credentials of the specified user profile. |

Console Tasks

Console tasks in System Center 2012 – Operations Manager run on the workstation where the Operations Console is running and uses the same credentials as the logged on user. The application that is run by the task must be installed on the workstation.

Console task



Console tasks are useful for running administration consoles or other tools that remotely access application components. These will usually be executable programs that run outside the Operations console. You can also run a command line utility and have the output displayed in a Console Task Output window.

For example, an administration console might require the name of the server that is running the application. The target for the task could be a class based on Windows Computer Role representing the application installation on the server. The task would only be available when an instance of the target class is selected. Because the class is hosted by Windows Computer, the PrincipalName property could be provided on the command line.

Create Console Task Wizard Options

When you run the Create Console Task wizard, you have to provide values for the options in the following tables. Each table represents a single page in the wizard.

General Properties

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Task Name | The name used for the task. This name is displayed in the Actions pane in the Operations console. |
| Description | Optional description of the task. |
| Task target | Target class of the task. The task will be displayed in the Actions pane when an instance of the target class is selected. Properties from the target object are available to use in the parameter of the task  You do not specify a target for Alert and Event command line tasks. They are available to all alerts and events regardless of the class that created them. |

Command Line

The following options are available on the Command Line page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Application | Path and name of the application to run. |
| Parameters | Parameters to add to the command line. This can be a combination of static text and variables for the properties of the target class or one of its parent classes.  If a variable used for a parameter could resolve to text containing a space, you should enclose the variable in quotations (""). If there are no quotations and the text includes a space, then it will be seen as multiple parameters. |
| Working directory | The default directory to use when the application is run. |
| Display output when this task is run | If selected, a Console Task Output window is opened when the task is run to display command line output. Make sure this option is selected if the application for the task is a utility that you would normally run from a command prompt. |

Command Line Tasks

Command line tasks run a command line application or batch file using a target class. They are listed in the Actions pane of the Operations console when an instance of the target class is selected. You can specify the path to the application and the working directory. You can also use $Target variables from the target class or one of its parents to be included on the command line.

The following procedure creates a console task to run the nslookup utility to resolve the name of a computer to an IP address on the local computer. In this example, you will pass the principle name of the target computer to the command line.

To create a command line task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Console Tasks, select Command line.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type NSLookup. This is the text that will be displayed in the Actions pane.  b. Click the Select button to open the Select Items to Target dialog box.  c. Select Windows Computer and click OK.  d. Click Next.  6. On the Command Line page, do the following:  a. In the Application box, type %windir%\system32\nslookup.exe.  b. Click the button to the right of the Parameters box and select Principal Name.  c. Ensure that the Display output when this task is run checkbox is selected. This option causes the Console Task Output window to be opened when the task is run.  d. Click Create. |

To test the command line task

|  |
| --- |
| 1. Select the Monitoring workspace.  2. Select the Windows Computers view.  3. Select a computer listed in the Windows Computers pane.  4. In the Windows Computer Tasks section of the Tasks pane, click NSLookup.  5. When you have inspected the results of the lookup, click Close. |

Alert Command Line Tasks

Alert command line tasks run a command line application or batch file based on an alert. They are listed in the Actions pane of the Operations console when an alert is selected. You can specify the path to the application and the working directory. You can also use properties from the selected alert on the command line.

The following procedure creates a command line task to run a custom application forwarding an alert to an external ticketing system. It runs a custom program that accepts the Ticket ID, Managed Object Name, and Owner as command line parameters. In this example, it is expected that the managed object name and owner would possibly include spaces, so they are enclosed in quotations.

To create an alert command line task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Console Tasks, select Alert command line.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type Forward Alert. This is the text that will be displayed in the Actions pane.  b. Click Next.  6. On the Command Line page, do the following:  a. In the Application box, type c:\ticketing\CreateTicket.exe.  b. Click the button to the right of the Parameters box and select Ticket ID.  c. Type a space after $Ticket ID$.  d. Click the button to the right of the Parameters box and select Managed Object Name.  e. Type quotations ("") around $Managed Object Name$ and a space after the second quotation.  f. Click the button to the right of the Parameters box and select Owner.  g. Type quotations ("") around $Owner$.  h. Verify that the final Parameters box looks like the following:  $Ticket ID$ "$Managed Object Name$" "$Owner$"  i. Ensure that the Display output when this task is run checkbox is not selected.  j. Click Create. |

Event Command Line Tasks

Event command line tasks run a command line application or batch file based on an event. They are listed in the Actions pane of the Operations console when an event is selected. You can specify the path to the application and the working directory. You can also use properties from the selected event on the command line.

The following procedure creates a command line task to run a custom application creating a ticket in an external ticketing system based on the selected event. It runs a custom program that accepts the Event Number, Managed Object Name, and Description as command line parameters. In this example, it is expected that the managed object name and description would possibly include spaces, so they are enclosed in quotations.

To create an event command line task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Console Tasks, select Event command line.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type Create Ticket. This is the text that will be displayed in the Actions pane.  b. Click Next.  6. On the Command Line page, do the following:  a. In the Application box, type c:\ticketing\CreateTicket.exe.  b. Click the button to the right of the Parameters box and select Number.  c. Type a space after $Number$.  d. Click the button to the right of the Parameters box and select Managed Object Name.  e. Type quotations ("") around $Managed Object Name$ and a space after the second quotation.  f. Click the button to the right of the Parameters box and select Description.  g. Type quotations ("") around $Description$.  h. Verify that the final Parameters box looks like the following:  $Ticket ID$ "$Managed Object Name$" "$Description$"  i. Ensure that the Display output when this task is run checkbox is not selected.  j. Click Create. |

See Also

[Tasks](#ze3b6e51e21d64a0196b0a80e9bd98e17)

[Agent Tasks](#zf6269883c45e4102a2bafa19739e06c7)

Agent Tasks

Agent tasks in System Center 2012 – Operations Manager are run on the agent computer where the target object is managed. An agent task can be a script or an executable program run from a command line. If an executable program is used, the application must be installed on the agent computer.

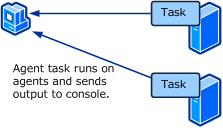
Agent tasks are useful for performing actions on the agent computer or for retrieving information for the user. They provide the following capabilities:

 Run a script or command locally on the agent computer without logging on to the computer interactively.

 Run a script or command on multiple agents with a single action.

 Run a script or command by using local user credentials with permissions not available to the user.

Agent Task



Credentials

Tasks run under the credentials of the Default Action Account on the agent computer. This account typically has sufficient privileges for accessing most application components, even if the user running the task does not have these user rights. If the task is required to perform an action requiring other credentials, such as accessing an external data source, then you can specify credentials when you run the task.

Output

Any output sent to the standard out stream (StdOut) from the script or command is provided to the user as Task Output in the Operations Console. Command line programs will typically output information to this stream. Scripts should output information by using commands such as WScript.Echo to provide this information.

Create Agent Task Wizard Options

When you run the Create Agent Task wizard, you have to provide values for the options in the following tables. Each table represents a single page in the wizard.

General Properties

The following options are available on the General Options page of the wizard.

|  |  |
| --- | --- |
| Option | Description |
| Task Name | The name used for the task. This name is displayed in the Actions pane in the Operations console. |
| Description | Optional description of the task. |
| Task target | Target class of the task. The task will be displayed in the Actions pane when an instance of the target class is selected. Properties from the target object are available to use in the parameter of the task  You do not specify a target for Alert and Event command line tasks. They are available to all alerts and events regardless of the class that created them. |

Command Line

The following options are available on the Command Line page of the wizard. This page is only available for a command line agent task.

|  |  |
| --- | --- |
| Option | Description |
| Application | Path and name of the application to run. |
| Parameters | Parameters to add to the command line. This can be a combination of static text and variables for the properties of the target class or one of its parent classes.  If a variable used for a parameter could resolve to text containing a space, you should enclose the variable in quotations (""). If there are no quotations and the text includes a space, then it will be seen as multiple parameters. |
| Working directory | The default directory to use when the application is run. |
| Timeout | The number of seconds that the application is allowed to run. If it has not completed within this time, the application will be ended and an error returned. |

UNIX/Linux Shell Command

The following options are available on the Shell Command Details page of the wizard. This page is only available for an UNIX/Linux Shell Command agent task.

|  |  |
| --- | --- |
| Option | Description |
| Command | The shell command to execute. This can be the full path to a program or script, a command, or a one-line sequence of multiple commands (using pipeline operators). |
| Run As Profile | Either the UNIX/Linux Action Account or UNIX/Linux Privileged Account profile. Select the profile that associates the required account credentials with the task target. The associated account will be used to execute the command. |
| Timeout (seconds) | The number of seconds that the command can run before the agent stops it. This prevents problem commands from running continuously and putting excess overhead on the agent computer. |

Script

The following options are available on the Script page of the wizard. This page is only available for a script agent task.

|  |  |
| --- | --- |
| Option | Description |
| File Name | Name of the script. Must end in a .vbs or .js extension depending on whether your script is written in VBScript or JScript. |
| Timeout | The number of seconds that the script can run before the agent stops it. This prevents problem scripts from running continuously and putting excess overhead on the agent computer. |
| Script | The body of the script. |
| Parameters | Click to provide values for any arguments in the script. For more information, see [Script Arguments](#z78). |

Creating Agent Tasks

Command Line Agent Tasks

Command line agent tasks run a command line application or batch file using a target class. They are listed in the Actions pane of the Operations console when an instance of the target class is selected. You can specify the path to the application and the working directory. This application must be installed on the agent computer when the task is run. You can also use $Target variables from the target class or one of its parents to be included on the command line. Any output from the application sent to the console are delivered back to the user in the Task Status dialog box in the Operations console when the task is run.

The following procedure creates a console task to run the netstat utility to list the ports that the agent computer is listening on.

To create a command line agent task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Agent Tasks, select Command line.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type Netstat. This is the text that will be displayed in the Actions pane.  b. Click the Select button to open the Select Items to Target dialog box.  c. Select Windows Computer and click OK.  d. Click Next.  6. On the Command Line page, do the following:  a. In the Application box, type %windir%\system32\netstat.exe.  b. Click Create. |

UNIX/Linux Shell Command Agent Tasks

UNIX/Linux Shell Command tasks run a command line application or script using a target class. They are listed in the Actions pane of the Operations console when an instance of the target class is selected. You can specify the path to a script or command, a command to run, or a one-line sequence of multiple commands (using pipeline operators). Any output from the application sent to the console are delivered back to the user in the Task Status dialog box in the Operations console when the task is run.

The following procedure creates a console task to run the netstat utility to list the ports that the agent computer is listening on.

To create an UNIX/Linux Shell Command agent task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Agent Tasks, select Run an UNIX/Linux Shell Command.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type Netstat. This is the text that will be displayed in the Actions pane.  b. Click the Select button to open the Select Items to Target dialog box.  c. Select UNIX/Linux Computer and click OK.  d. Click Next.  6. On the Shell Command Details page, do the following:  a. In the Command box, type netstat.  b. Select the Run As profile to use.  c. Input the task timeout in seconds.  d. Click Create. |

Script Agent Tasks

Script agent tasks run a Windows script using a target class. They are listed in the Actions pane of the Operations console when an instance of the target class is selected. The script can perform some action or it can collect information that is delivered back to the user in the Task Status dialog box. To return information to the user, you can use any script command that will display information on the command line. For VBScript, this will typically be the WScript.Echo command.

The following procedure creates an agent script task to reboot the target computer. It accepts two parameters. The first is the name of the computer that it retrieves from the Principal Name property of the target. The second is a flag that specifies that the reboot should be performed. This value defaults to false, and the user must change it to true when they run the task or the reboot will not be performed. This provides an additional safety to the task to prevent an operator from accidentally performing a reboot.

To create a command line agent task

|  |
| --- |
| 1. Select the Authoring workspace.  2. In the Authoring pane, expand Management Pack Objects.  3. Right click Tasks and select Create a New Task to open the Create Task Wizard.  4. On the Task Type page, do the following:  a. Under Agent Tasks, select Run a script.  b. In the Select destination management pack dropdown, select the management pack file to store the task. For more information about management packs, see [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  c. Click Next.  5. On the General Properties page, do the following:  a. Under Task Name, type Reboot Computer. This is the text that will be displayed in the Actions pane.  b. Click the Select button to open the Select Items to Target dialog box.  c. Select Windows Computer and click OK.  d. Click Next.  6. On the Script page, do the following:  a. In the File Name box, type RebootComputer.vbs.  b. Leave the default of 1 minute for the Timeout.  c. Copy the following script and past into the Script box.  sComputer = WScript.Arguments(0)  bConfirmFlag = cbool(WScript.Arguments(1))  Set colOS = GetObject("winmgmts:{impersonationLevel=impersonate,(Shutdown)}//" & sComputer).ExecQuery("select \* from Win32\_OperatingSystem where Primary=true")  If bConfirmFlag Then  For each objOS in colOS  objOS.Reboot()  Next  Else  WScript.Echo "Confirm flag set to false. Computer will not be rebooted."  End If  d. Click Parameters to open the Parameters dialog box.  e. Click Target and then select Principal Name.  f. Type a space after the computer name variable and then type false. The final parameter line should look similar to the following:  $Target/Property[Type="MicrosoftWindowsLibrary7585000!Microsoft.Windows.Computer"]/PrincipalName$ false  g. Click OK.  h. Click Create. |

See Also

[Tasks](#ze3b6e51e21d64a0196b0a80e9bd98e17)

[Console Tasks](#z571118fba58a43a9a13a5abd1fde2f7d)

Distributed Applications

A Distributed Application in Operations Manager is an application that is comprised of multiple objects. These objects may be defined by different management packs and may be managed on the same agent or on different agents. The purpose of the distributed application is to provide an overall health for an application that is comprised of different objects.

Distributed applications do not provide any additional monitoring for the objects in an application. Instead, they include objects that are already being monitored. The value of the distributed application is to provide a relationship between the health of objects that are part of a single application.

You can create distributed applications using the Distributed Applications Designer in the Operations console.

Distributed Application Designer

The Distributed Application Designer provides you with the ability to create a distributed application in a graphical environment with minimal knowledge of the underlying management pack elements that are created. There are some limitations to the tool though as follows:

 The component groups can only have explicit members, which means that you cannot populate them dynamically. For example, you may have a distributed application with a set of web sites. You install an additional web server with a site that should be included. You would have to edit the distributed application and manually add the new site.

 You cannot create multiple levels of health rollup. The health of the application will be the worst health of any of the component groups regardless of the relationships that have been created for them.

You can create a distributed application without these limitations by using the [Visual Management Pack Designer](#z2).

Contents of Distributed Applications

Objects

A distributed application must include one or more objects in order to be useful. Any object discovered by different management packs installed in the management group can be used in a distributed application. This might come from a management pack installed from the catalog or one that you have created on your own. These can be objects created by different monitoring wizards as discussed in [Management Pack Templates](#z7f7e6aa7678e4436841b0e097ef4e4ca).

Component Groups

A component group can contain any number of objects, and any object added to the distributed application must be contained in a component group. When you create the component group, you specify one or more classes that the group can contain. Only objects that are instances of these classes may be added to the particular group. If you specify All Objects then any objects in the management group can be included in the component group.

If you want to limit the objects that can be included in the component group, then you should select the Object(s) if the following type(s) and then select one or more classes from the class tree. The tree will contain all of the classes in the management group which are provided by all the management packs currently installed.

The dialog box arranges classes in a tree according to their base classes. You can read more about base classes at [Base Classes](#z12). If you select a class, then each of its base classes will also be selected. This allows you to select a single class such as Computer Role that is often used as a base class and automatically select all of its base classes.

Relationships

Relationships can be drawn between component groups to represent some relation between different kinds of objects. Health is not rolled up between the component groups, but the relationship is indicated by lines between the groups.

Distributed Application Templates

Templates allow you to start a new distributed application with a predefined set of component groups. You need to add objects to these component groups and can modify the component groups and add additional component groups as you require. You can save time in creating a distributed application by selecting a template that most closely resembles your requirements. If you want to create an empty distributed application and manually add all your own component groups, then select the Blank template.

The following table lists the available templates:

|  |  |  |  |
| --- | --- | --- | --- |
| Template | Description | Container Groups | Contained Classes |
| .NET 3-Tier Application | Brings together objects and monitoring data from synthetic transactions with data from Application Performance Monitoring | **** [Application name] Client Perspective  **** [Application name] Presentation Tier  **** [Application name] Business Tier  **** [Application name] Data Tier | ****  Perspective  **** ASP.NET application  **** .NET application component  **** Database |
| Line of Business Web Application | Component groups common to a web application | **** Web Sites  **** Databases | **** Web Site  **** Database |
| Messaging | Component groups common to messaging services | **** Messaging Clients  **** Storage  **** Messaging Components  **** Directory Services  **** Network Services  **** Physical Network | **** Perspective  ****  **** Logical Hardware Component  **** Computer Role  **** Computer Role  **** Network Device |
| Blank | Empty distributed application with no component groups | None | None |

Viewing Distributed Applications

Each distributed application will be listed in the Distributed Application state view in the Monitoring pane of the Operations console. The state of the distributed application will be the worst state of any of the objects that it contains. You can launch any of the other kinds of the view by right-clicking on the distributed application and selecting the view that you want. Each view will include data for all of the objects contained in the distributed application.

Creating and Modifying Distributed Applications

To create a distributed application

|  |
| --- |
| 1. If you don’t have a management pack for the application that you are monitoring, create one using the process in [Selecting a Management Pack File](#z8e8c3975e6d542c3aa3f1f666986688c).  2. In the Operations console, select the Authoring workspace.  3. Right-click Distributed Applications and select Create a new distributed application.  4. In the Name box, type a name for the distributed application. This name will appear in the Monitoring workspace of the Operations console.  5. In the Template box, select the template for the starting point of the distributed application. See [Distributed Application Templates](#z80) for information on the available templates.  6. Select the management pack that you created in step 1.  7. Click OK. |

To edit an existing distributed application

|  |
| --- |
| 1. In the Operations console, select the Authoring workspace.  2. Select Distributed Applications.  3. In the Distributed Applications pane, right-click the distributed application you want to edit and select Edit. |

To create a component group

|  |
| --- |
| 1. With the distributed application open, click Add Component.  Note  The first time that you create a component group since opening the Operations console, it may take several seconds to open the Create a New Component Group dialog box since the list of classes must be cached. The amount of time that this takes will depend on the number of classes in your management group.  2. In the Name your component group text box, provide a name for the component group. This is the name that will appear in the diagram view and the Health Explorer for the distributed application.  3. If the component group should be able to contain any type of object, then select All Objects. If you want to specify one or more types that the component group should be able to contain, then select Objects of the following type.  4. Select one or more classes to allow objects of that type to be included in the management group.  5. Click OK when you have selected the classes.  Note  You may receive a message saying that the allowed limit was reached while making a new object type button visible. This means that no more object selection panes can be added to left side of the Distributed Application Designer. Either select Leave the new object type not visible to not create a new object selection panel for the current component group or select Replace the selected visible Object Type button with a new one and select one of the object types in the list to close. You can reopen the selection pane by selecting it in the Organize Object Types pane. |

To add an object to the distributed application

|  |
| --- |
| 1. Ensure that a component group is created that allows the type of object that you want to add.  2. If a selection pane is not open for the type of object you want to add, click Organize Object Types and then select the type of object you want to add.  3. Select the type of object you want to add in the Object Picker. This should display a list of all objects of the selected type.  4. Click and drag one or more of the objects into a component group. Note that you will only be able to add the objects to a component group that will accept objects of that type. |

See Also

[Understanding Classes and Objects](#za7a6764a82df4c628298695a8872fd3f)

360 .NET Application Monitoring Dashboards in System Center 2012 SP1

The information in this topic applies only to System Center 2012 SP1.

System Center 2012 – Operations Manager can show you different perspectives of application health in one place—360 .NET Application Monitoring Dashboards. The 360 .NET Application Monitoring Dashboards display information from Global Service Monitor, .NET Application Performance Monitoring, and Web Application Availability Monitoring to provide a summary of health and key metrics for 3-tier applications in a single view. The 360 .NET Application Monitoring Dashboards show where an application is unhealthy and provide a launch point for Application Summary Dashboards that highlight component-level issues, which are helpful when troubleshooting.

The 360 .NET Application Monitoring Dashboards display data from powerful monitoring tools. .NET Application Performance Monitoring looks deep into the application to get details that can help you pinpoint solutions from server- and client-side perspectives. Web Application Availability monitoring in Operations Manager monitors internal synthetic transactions. Global Service Monitor monitors the availability of applications from an outside location, measuring availability from where the user is.

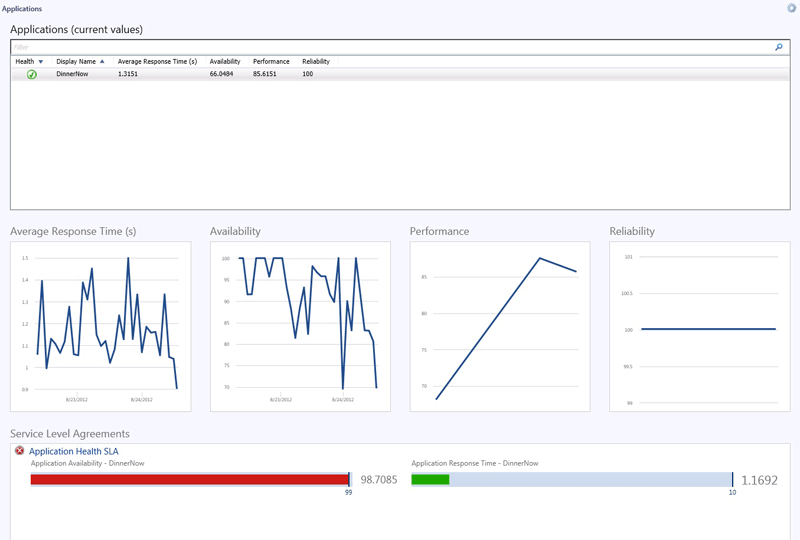
The Applications Dashboard

The Applications Dashboard gives you a holistic view of all applications modeled through a Distributed Application (DA) and help answer these questions:

 Are my applications healthy? This is provided by the state view that shows the DA health as measured by availability from Global Service Monitor.

 Do the high-level metrics indicate a problem? Four metric scores provide this information, showing Average response time (Global Service Monitor), Availability (Global Service Monitor), Performance Exceptions (.NET Application Performance Monitoring), and Reliability (exception events from .NET Application Performance Monitoring.)

 Is the application meeting SLAs? Once you select a DA, the lower part of the dashboard retrieves all SLAs defined for that DA, and show a state for that SLA.



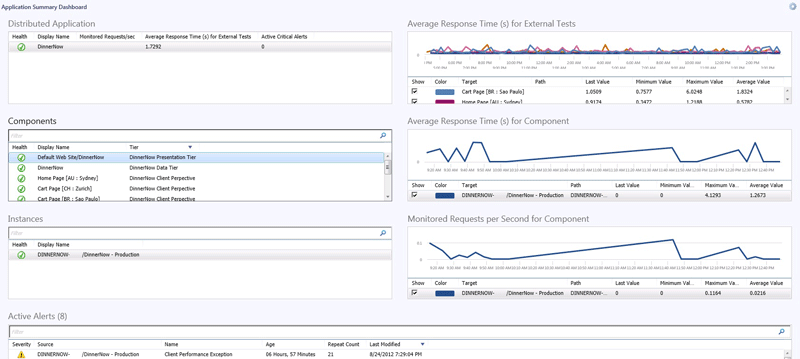
The Applications Dashboard highlights the following:

|  |  |
| --- | --- |
| Key application metric or action | Type of monitoring used |
| Availability | Web Application Availability Monitoring and Global Service Monitor |
| Reliability | .NET Application Performance Monitoring |
| Performance | .NET Application Performance Monitoring and Global Service Monitor |

The Application Summary Dashboard

The Application Summary Dashboard shows components and instances of applications, performance data, and active alerts. The data are not aggregated, so the Application Summary Dashboard is a useful tool for troubleshooting.

When you select an application in the Applications Dashboard, Application Summary Dashboard appears in the right rail. To open it, click Application Summary Dashboard.



The Applications Summary Dashboard highlights the following:

|  |  |
| --- | --- |
| Key application metric or action | Type of monitoring used |
| Distributed Application | The distributed application that you selected on the Applications Dashboard |
| Components | Components discovered by .NET Application Performance Monitoring and Global Service Monitor that are part of the selected distributed application. When you select a component, the related data displays in the Average Response Time (s) for Component and Monitored Requests per Second for Component charts.  Note  Not all components have the associated data in these two charts—only the components discovered by .NET Application Performance Monitoring. |
| Instances | The instances of each component. For example, you might have several instances of a website running on several computers. |
| Active Alerts | Active alerts for the selected distributed application |
| Average Response Time (s) for External Tests | Average response time in seconds for the monitoring tests that you have configured in Global Service Monitor for the selected distributed application |
| Average Response Time (s) for Component | Average response time in seconds for the selected component as configured by .NET Application Performance Monitoring |
| Monitored Requests per Second for Component | Monitored requests per second for the selected component as configured by .NET Application Performance Monitoring |

See Also

[Before You Begin Using 360 .NET Application Monitoring Dashboards in System Center 2012 SP1](#zb563dc5166c941198f6a17429d639304)

[How to Configure 360 .NET Application Monitoring Dashboards in System Center 2012 SP1](#z04b25f28302e46b38948f34656918764)

[How to Use the 360 .NET Application Monitoring Dashboards in System Center 2012 SP1](#z640a17c926454992aa2c38619bbdc57d)

Before You Begin Using 360 .NET Application Monitoring Dashboards in System Center 2012 SP1

The information in this topic applies only to System Center 2012 SP1.

Before you begin using 360 .NET Application Monitoring Dashboards, you must configure application monitoring for the applications you want to monitor.

Configure Application Monitoring

Before you begin using 360 .NET Application Monitoring Dashboards, you must configure .NET Application Performance Monitoring, Web Application Availability Monitoring, and Global Service Monitor for the applications you want to monitor. For more information, see:

 [.NET Application Performance Monitoring Template](#z2166e40acc7d4534ad8e73a51ac2bf99)

 [Web Application Availability Monitoring Template](#z62d3b2b3f4ff4fe180553078179c1b1e)

 Global Service Monitor documentation is on Connect. To get the documentation and get started, go to <http://connect.microsoft.com> and download Global Service Monitor Installer (Windows Installer).

Important

Before configuring the 360 .NET Application Monitoring Dashboards, make sure to allow enough time for all of the applications you want to monitor to be discovered.

How to Configure 360 .NET Application Monitoring Dashboards in System Center 2012 SP1

The information in this topic applies only to System Center 2012 SP1.

To configure 360 .NET Application Monitoring Dashboards

|  |
| --- |
| 1. In Operations Manager, click the Authoring button.  2. Right-click Distributed Applications and select Create a new distributed application. The Distributed Application Designer opens.  3. In the Distributed Application Designer, choose a name for your distributed application and enter a description (optional). In the Choose Distributed Application Template, select the .NET 3 Tier Application template. In the Save to a Management Pack section, select an existing management pack or create a new management pack where your distributed application and its components will be saved. Click OK.  4. In the Distributed Application Designer, you can drag and drop the discovered application instances and components you want to monitor and view on the Application Summary Dashboard. Headings in the Objects pane map to the boxes. The items you drag in should be in the context of what you previously configured for.NET Application Performance Monitoring, Web Application Availability, and Global Service Monitor.    To select components and instances of the distributed application that you want to view in the Application Summary Dashboard, in Object Types, click an object type, select the instances you want display in the Application Summary Dashboard, and then drag them to the matching component group in the main flow.  Note  If you configured applications to monitor using 360 .NET Application Monitoring Dashboards, but do not see them in the Distributed Application Designer, you might need to allow more time for all of the applications you want to monitor to be discovered.  5. For more information about distributed applications and the Distributed Application Designer, see [Distributed Applications](#z9deed7cf5546485f9276a1e4ad9dd928). |

To change thresholds for SLAs

|  |
| --- |
| 1. To change thresholds for SLAs, in Operations Manager click the Authoring button, click Management Pack Objects, click Service Level Tracking, and then double-click Application Health SLA.  2. In the Service Level Tracking wizard, on the Service Level Objectives page, you can add, edit, or remove service level objectives, which define the performance thresholds or the states that you want to track for the selected targeted class, objects, or group. When you are done, click Finish.  Note  You can also add SLAs and SLOs and they will display in the Application Dashboard automatically. |

To set the time range for data displayed in the dashboards

|  |
| --- |
| 1. To set a time range for the data displayed in the Applications Dashboard, click the round icon in the upper right corner and click Personalize. Set the time range for data collections and then click Finish.  2. To set a time range for the data displayed in the Applications Summary Dashboard, click the round icon in the upper right corner and click Personalize. Set the time range for data collections and then click Finish. |

To personalize the Application Summary Dashboard

|  |
| --- |
| 1. To personalize the Application Summary Dashboard, hover over the section you want to modify, click the round icon above its upper right corner, and then click Personalize.  Note  You can personalize the Average Response Time(s) for Externals Tests, Average Response Time(s) for Component, Monitored Requests per Second for Component, and the Active Alerts sections. The Distributed Application, Components, and Instances sections cannot be personalized.  2. Select the display or chart preferences for the information you want the dashboard to display, and then click Finish. |

To select locations to display on the Average Response Time(s) for External Tests dashboard

|  |
| --- |
| 1. On the Application Summary Dashboard, in the Average Response Time(s) for External Tests section, the application components and locations you have configured to monitor from are listed and displayed on the chart. To change which application component and locations are displayed on the chart, select only the application component locations you want to see.  2. The chart will update automatically. |

To synchronize active alerts in the Application Summary Dashboard with Team Foundation Server (TFS) work items

|  |
| --- |
| 1. Import the Team Foundation Server (TFS) Work Item Management Pack into Operations Manager. This management pack synchronizes Operations Manager alerts with TFS work items so alerts from Operations Manager can be sent directly to development where they can view and track it in Visual Studio and remain synchronized with the alert you see in Operations Manager.  2. When the management pack is imported, assign an alert to engineering and this will synchronize the alert with TFS automatically. |

How to Use the 360 .NET Application Monitoring Dashboards in System Center 2012 SP1

The information in this topic applies only to System Center 2012 SP1.

To open and use the Application Dashboard

|  |
| --- |
| 1. In Operations Manager, click the Monitoring button, click Application Monitoring, and then click Applications.  2. To open the Application Dashboard, click the application you want to see details about.  The 360 .NET Application Monitoring Dashboards show:   A list of all applications configured to appear in the 360 .NET Application Monitoring Dashboards   The current state of each 360 distributed application   Aggregated Average Transaction Response Time (in seconds) Performance Data from Global Service Monitor   Availability percentage for the last 24 hours represented as a score   Aggregated % Performance events for the last 24 hours represented as a score. (100 is a perfect score with no performance exceptions. This will be reduced as the % performance events are increased.)   Aggregated % Exception events represented as a score. (100 is a perfect score with no exception events. This number will be reduced as the % exception events are increased.)   The history of these counters, each in a performance graph, for the selected interval   SLA information. The default SLA thresholds can be changed. You can also add your own SLAs as long as they are targeted at the distributed application. These SLAs will also appear in the dashboard.  You can also change the interval shown in the dashboard by clicking Personalize. The default is 24 hours.  Note  Information, such as average response, availability, performance, and reliability in the Applications Dashboard is aggregated. Aggregate performance counters provide value since the application owner typically wants to know how their website is performing, not how the individual instances of the website are performing. Instance-level detail is not required until troubleshooting begins. |

To open and use the Application Summary Dashboard

|  |
| --- |
| 1. The Application Summary Dashboard shows components and instances of applications, performance data, and active alerts. The data are not aggregated, so the Application Summary Dashboard is a useful tool for troubleshooting.  To view an Application Summary Dashboard, click an application in the Applications Dashboard. Application Summary Dashboard appears in the right pane.  2. To open the Application Summary Dashboard, click Application Summary Dashboard. |