Foglight™ for Microsoft .NET 5.9.10

Installation and Configuration Guide

Copyright© 2017 Dell Inc. All rights reserved.

This product is protected by U.S. and international copyright and intellectual property laws. Dell™, the Dell logo, Foglight, IntelliProfile, NetVault, OpenManage, PerformaSure, PowerEdge, Tag and Follow, and Toad are trademarks of Dell Inc. in the United States and/or other jurisdictions. “Apache HTTP Server”, Apache, “Apache Tomcat” and “Tomcat” are trademarks of the Apache Software Foundation. Google is a registered trademark of Google Inc. Android, Chrome, Google Play, and Nexus are trademarks of Google Inc. Red Hat, JBoss, the JBoss logo, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries. CentOS is a trademark of Red Hat, Inc. in the U.S. and other countries. Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Microsoft, .NET, Active Directory, Internet Explorer, Hyper-V, Office 365, SharePoint, Silverlight, SQL Server, Visual Basic, Windows, Windows Vista and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX, IBM, PowerPC, PowerVM, and WebSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java, Oracle, Oracle Solaris, PeopleSoft, Siebel, Sun, WebLogic, and ZFS are trademarks or registered trademarks of Oracle and/or its affiliates in the United States and other countries. SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries. Products bearing the SPARC trademarks are based on an architecture developed by Oracle Corporation. OpenLDAP is a registered trademark of the OpenLDAP Foundation. HP is a registered trademark that belongs to Hewlett-Packard Development Company, L.P. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Novell and eDirectory are registered trademarks of Novell, Inc., in the United States and other countries. VMware, ESX, ESXi, vSphere, vCenter, vMotion, and vCloud Director are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Sybase is a registered trademark of Sybase, Inc. The X Window System and UNIX are registered trademarks of The Open Group. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. “Eclipse”, “Eclipse Foundation Member”, “EclipseCon”, “Eclipse Summit”, “Built on Eclipse”, “Eclipse Ready”, “Eclipse Incubation”, and “Eclipse Proposals” are trademarks of Eclipse Foundation, Inc. IOS is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Apple, iPad, iPhone, Mac OS, Safari, Swift, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Ubuntu is a registered trademark of Canonical Ltd. Symantec and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. OpenSUSE, SUSE, and YAST are registered trademarks of SUSE LCC in the United States and other countries. Citrix, AppFlow, NetScaler, XenApp, and XenDesktop are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. AlertSite and DéjàClick are either trademarks or registered trademarks of Boca Internet Technologies, Inc. Samsung, Galaxy S, and Galaxy Note are registered trademarks of Samsung Electronics America, Inc. and/or its related entities. MOTOROLA is a registered trademarks of Motorola Trademark Holdings, LLC. The Trademark BlackBerry Bold is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. Dell is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited. Ixia and the Ixia four-petal logo are registered trademarks or trademarks of Ixia. Opera, Opera Mini, and the O logo are trademarks of Opera Software ASA. Tevron, the Tevron logo, and CitraTest are registered trademarks of Tevron, LLC. PostgreSQL is a registered trademark of the PostgreSQL Global Development Group. MariaDB is a trademark or registered trademark of MariaDB Corporation Ab in the European Union and United States of America and/or other countries. Vormetric is a registered trademark of Vormetric, Inc. Intel, Itanium, Pentium, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Debian is a registered trademark of Software in the Public Interest, Inc. OpenStack is a trademark of the OpenStack Foundation. Amazon Web Services, the “Powered by Amazon Web Services” logo, and “Amazon RDS” are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. Infobright, Infobright Community Edition and Infobright Enterprise Edition are trademarks of Infobright Inc. POLYCOM®, RealPresence® Collaboration Server, and RMX® are registered trademarks of Polycom, Inc. All other marks and names mentioned herein may be trademarks of their respective companies.

Legend

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight for Microsoft .NET Installation and Configuration Guide
Updated - January 2017
Foglight Version - 5.7.5.7
Cartridge Version - 5.9.10

Contents

Installing and Configuring Foglight for Microsoft .NET 4
System requirements 4
Understanding the .NET agent architecture 5
Supported platforms and versions for monitored applications 5
Nexus requirements 5
Adjusting the Agent Manager settings for the Nexus 6
Installing Foglight for Microsoft .NET 7
Creating .NET Agent instances 8
Creating a .NET transaction and performance counters monitoring agent 8
Configuring a remote host for .NET performance counters monitoring 11
Creating a .NET performance-counters-only monitoring agent 11
Reusing .NET agent integration configurations 13
Reusing .NET transactions and performance counters integration 13
Reusing a .NET performance-counters-only integration configuration 15
Viewing .NET agent task history 16
Upgrading .NET agents 16

Managing .NET Agent Configurations 18
Managing .NET installations 18
Uninstalling .NET installations 18
Viewing .NET installation properties 19
Managing .NET transaction agent configurations 19
Creating .NET transaction agent configurations 19
Viewing and editing .NET transaction agent configurations 20
Filtering application pools monitoring 21
Managing .NET agent integration configurations 22
Viewing and editing .NET integration configurations 22
Editing IIS .NET integration configurations 23
Editing Windows Services .NET integration configurations 24
Editing standalone applications .NET integration configurations 24
Editing .NET thread metrics integration configurations 25
Switching .NET agent configurations 25
Understanding the Nexus and agent-Nexus connections for .NET agents 26
Managing .NET agent Nexus connections 26
Updating the Nexus connection for a target application server 27
Switching a Nexus connection 28
Managing collection schedules 28
Creating Collection Schedules 29
Assigning custom collection schedules to an agent 29
Increasing WMI polling memory values 30
Preventing WMI quota violations 30

About Dell 31

1

Installing and Configuring Foglight for Microsoft .NET

Many .NET system developers and production specialists need a .NET analysis tool because their application is not meeting specific performance requirements. Foglight for Microsoft .NET provides 24x7 monitoring to help you detect and resolve performance issues in your application server infrastructure and in the applications running in that infrastructure.

Foglight for Microsoft .NET monitors the health of your web servers and application servers, and their host operating systems, in your integrated environment. The .NET agent collects the data and sends it to the Nexus (a central analytical engine). The Nexus correlates the data and sends it to the Foglight Management Server. For more information, see Instrumented agents on page 4.

Foglight for Microsoft .NET includes a set of monitoring and configuration dashboards, as well as several rules that notify application and system administrators about performance problems. This guide discusses the dashboards in the context of the tasks you can perform from them.

These topics contain information about the system requirements and takes you through the installation procedure, step-by-step.

Instrumented agents

An instrumented agent collects performance information from .NET domains and passes this data to the Foglight for Microsoft .NET Management Server through the Nexus and the Agent Manager. At the application level, service request agents use bytecode instrumentation to measure method-level performance information. At the server level, agents monitor the application server metrics through vendor-provided interfaces. At the .NET run-time level, agents monitor various metrics; at the operating system level, they monitor platform-level metrics. You can create agents to monitor servers after you install the cartridge file on the Management Server.

For information on installing and configuring instrumented agents, see the following sections:

- System requirements on page 4
- Adjusting the Agent Manager settings for the Nexus on page 6
- Installing Foglight for Microsoft .NET on page 7
- Creating .NET Agent instances on page 8
- Upgrading .NET agents on page 16

System requirements

Before installing Foglight for Microsoft .NET, ensure that you have the required environment. Review the following resources:

- *Foglight System Requirements and Platform Support Guide*
- Understanding the .NET agent architecture
- Supported platforms and versions for monitored applications
- Nexus requirements

- Adjusting the Agent Manager settings for the Nexus.

Understanding the .NET agent architecture

There are two configurations for .NET agent:

1 The **Performance Counters** agent. This agent collects periodic metric data and requires:

- A Foglight Agent Manager (FglAM) installed on the host.
- IIS Management Scripts and Tools must be installed on the monitored host in order for the .NET agent to collect WMI metrics.

2 The **Transactions and Performance Counters** agent. This agent collects both periodic metric data and request trace data and requires:

- The same configuration as specified for the Performance Counters agent.

The installer also installs the *AgentHost.exe* Windows® service that acts as a controller for the .NET agents embedded in the individual application pools on the monitored system.

- *AgentHost. exe* requires .NET 4.0 installed on the host, but .NET 4.0 is not required to be part of your monitored application runtime environment.

Supported platforms and versions for monitored applications

The Foglight for Microsoft .NET agents monitor applications on the following platforms and of the stated versions:

Table 1. Supported systems for .NET Framework versions

.NET Framework Version	Windows® XP SP1+ (ia32, x86-64)	Windows Server® 2003 (ia32, x86-64)	Windows Server 2008 (ia32, x86-64)	Windows Server 2008 R2 (ia32, x86-64)	Windows Server 2012 (x86-64)	Windows Server 2012 R2 (x86-64)	Windows Server 2016 (x86-64)
	IIS 6.0	IIS 6.0	IIS 7.0	IIS 7.5	IIS 8.0	IIS 8.5	IIS 10.0
2.0, SP1, SP2	Y	Y	Y	Y	Y	Y	Y
3.0, SP1, SP2	Y	Y	Y	Y	Y	Y	Y
3.5, SP1	Y	Y	Y	Y	Y	Y	Y
4.0	Y	Y	Y	Y	Y	Y	Y
4.5, 4.5.[1-2]	N	N	Y	Y	Y	Y	Y
4.6	N	N	Y	Y	Y	Y	Y
4.6.[1-2]	N	N	N	Y	Y	Y	Y

For information about supported browsers, see the *Foglight System Requirements and Platform Support Guide*.

Nexus requirements

The Nexus runs on Foglight Agent Manager version 5.6.7.2 or later.

If you are using the .NET Transaction agent, ensure that the following requirements for the Nexus are met.

> **IMPORTANT:** The Agent Manager heap requires at least 1Gb of memory for the Nexus. Depending on the monitored application, this heap requirement may increase to as much as 4Gb.

The Nexus requires Oracle® (Sun) Java SE JDK 1.6, 1.7, or 1.8.

The Nexus is supported on the following OS platforms:

Table 2. Supported OS platforms for the Nexus

OS version	Architecture
Oracle Solaris® 8, 9, 10	SPARC®
Oracle Solaris 10, 11	SPARC, x86-64
Windows® XP SP1+, Server 2003, Server 2008	ia32, x86-64
Windows Server® 2008 R2, Server 2012, Server 2012 R2, Server 2012 R2 Core, Server 2016	x86-64
Red Hat® AS/ES/WS 4, 5, 6, 7	ia32, x86-64
SUSE® 9, 10, 11, 12	ia32, x86-64

Final platform support notice

Support for the following platforms will be discontinued in the next major release:

- Solaris 8 and 9
- All 32-bit versions of Oracle Solaris® SPARC®
- Windows® XP SP1+
- Red Hat® AS/ES/WS 4.x
- SUSE® 9

Adjusting the Agent Manager settings for the Nexus

If you intend to use the default Nexus running in the embedded Foglight for Microsoft .NET Agent Manager, adjust the Agent Manager disk cache settings before installing and configuring Foglight for Microsoft .NET. The default Nexus is created as part of the installation process.

The Agent Manager accumulates messages that are destined to be sent either upstream or downstream in queues between connections. This prevents messages from getting lost in the event of a disconnection.

Increasing the Agent Manager disk cache size ensures that data submissions from the Java EE agent or the .NET agent are not discarded. See the documentation in the *fglam.config.xml* file for detailed descriptions of the options.

> **NOTE:** This procedure is recommended for all installations.

To adjust the Agent Manager settings for use with the Nexus:

1. Navigate to your Agent Manager `config` folder. For example: `C:\Quest_Software\Foglight_Agent_Manager\state\default\config`.
2. Open the *vm.config* file in an editor.

> **IMPORTANT:** In Agent Manager versions 5.6.10 and later, this file is called baseline.jvmargs.config.

3 Uncomment and set the following setting to:

```
vmparameter.0 = "-Xmx1024m";
```

4 Save and close the *.config* file.

5 In the same config directory, open the *fglam.config.xml* file in an editor.

6 Edit the `config:queue-sizes` and `max-disk-space` settings as follows:

```
<config:queue-sizes>
    <config:upstream max-queue-size="500" max-disk-space="100000" max-batch-size="500" allow-runtime-change="false"/>
    <config:upstream-verified max-queue-size="250" max-disk-space="50000" max-batch-size="250" allow-runtime-change="false"/>
    <config:downstream max-queue-size="500" max-disk-space="1024" max-batch-size="500" allow-runtime-change="false"/>
</config:queue-sizes>
```

This sets the `max-disk-cache` size to 100MB for the upstream and upstream-verified queues (normal and verified submissions).

7 Save and close the *fglam.config.xml* file.

8 Restart the Agent Manager.

If the Nexus is running in a non-embedded Agent Manager, the Agent Manager service must be restarted. This is the more common case.

If the Nexus is running in the embedded Agent Manager (for example, if you are using the default Nexus), do one of the following:

- Restart the Management Server.

Or

- Go to `<server>/jmx-console (for example, http://torrdv288:8080/jmx-console/)` and click: `service=EmbeddedFglAm`.

  Invoke the `Stop()` operation, and then invoke the `start()` operation.

If you still find warnings in the Agent Manager log about the disk cache being too small, increase the `max-disk-space` values for the upstream and upstream-verified queues.

Next step

- After you have adjusted the settings and restarted the Agent Manager, continue with Installing Foglight for Microsoft .NET.

Installing Foglight for Microsoft .NET

Prerequisite

- Adjusting the Agent Manager settings for the Nexus on page 6

Foglight for Microsoft .NET is distributed in a cartridge (*.car*) file, *ApplicationServers-DotNET-<version>.car*. Installing and enabling the cartridge file on the Management Server adds the capabilities for monitoring Microsoft .NET systems to your Foglight for Microsoft .NET environment.

The installation process is common to all Foglight cartridges. For details, see Installing Foglight cartridges in the *Administration and Configuration Help*.

> **TIP:** Unlike with previous versions, you do not need to manually deploy the agent package to the hosts. The agent creation process now manages this step.

Creating .NET Agent instances

Foglight for Microsoft .NET uses the .NET agent to collect information from monitored hosts. Creating a .NET agent instance creates the agent process (either locally, or on the monitored host).

Use the Application Servers Administration dashboard to create .NET agent instances.

> **NOTE:** This dashboard is common to Foglight for Microsoft .NET and Foglight for Java EE Technologies, versions 5.9.1 and later.

To create a .NET agent:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.

   The Application Server Administration dashboard opens in the display area.

   > **IMPORTANT:** The options available on this dashboard depend on the application server monitoring capabilities that you have installed (for example, Java Administration is only available if you have installed Foglight for Java EE Technologies).

2. In the Monitoring Agents section, click **Setup agents...**
3. In the list that opens, click **.NET**. The .NET Setup wizard opens.
4. On the Setup Options page, select the type of .NET agent to create, depending on whether you want to monitor **Transactions and performance counters** or **Performance counters only.**

   Transaction monitoring - Install the agent on the monitored host. This means that you must have an Agent Manager deployed to the monitored host. For more information, see Creating a .NET transaction and performance counters monitoring agent on page 8.

   Performance-counter-only monitoring - Install the agent either on the monitored host or on a remote host. For remote hosts, review the topic Configuring a remote host for .NET performance counters monitoring on page 11 before continuing with Creating a .NET performance-counters-only monitoring agent on page 11.

Creating a .NET transaction and performance counters monitoring agent

Before you begin

Ensure that you have installed and activated a compatible version of the Foglight Agent Manager on the monitored host. If you are not familiar with this process, see the *Foglight Agent Manager Guide*.

> **NOTE:** You cannot create a transaction and performance counters monitoring agent on a host that already has an existing .NET performance counters monitoring agent installed locally.

This procedure continues from the end of the procedure To create a .NET agent: on page 8.

To create a Transaction and Performance Counters Monitoring Agent:

1. On the Setup Options page, select **Transactions and performance counters** and click **Next**.
2. On the Monitored Hosts page, select one or more hosts for the .NET agent.

> **IMPORTANT:** By default, only eligible hosts (hosts that have an active, compatible Agent Manager and that are not being monitored by a performance counters agent) are shown. To view all hosts, click Show eligible hosts only. The Monitored Hosts page refreshes and all hosts are shown.

Select hosts from the list and click **Next**.

3 On the Integration Configuration page, select whether you want to create a configuration or reuse an existing one. This configuration controls whether the IIS is integrated, and specifies which Windows® Services and standalone applications are integrated.

To learn about reusing an existing configuration, see Reusing .NET transactions and performance counters integration on page 13.

For this example, select **Create new configuration**, and click **Next**.

4 On the New Integration Configuration page, specify integration targets for the agent. These targets can include IIS, Windows Services, and standalone applications.

- **Integrate with IIS** is enabled by default. If you do not want to integrate with IIS, clear this check box.
- To integrate with a Windows Service, click **Add Windows Service**.

  A row appears in the list, with the Process Type specified as Windows Service.

  Click the **Name** box, and type the name of the Windows Service. This must be the service name, not the display name.
- To integrate with a standalone application, click **Add standalone application**.

  A row appears in the list, with the Process Type specified as Standalone Application.

  Click the **Name** box, and type the full path to the executable.

After you have specified all the targets for integration, click **Next**.

5 On the Transaction Agent — Nexus Connection page, select or create the connection that the agent should use to connect to the Nexus.

> **TIP:** For more information about the Nexus, see Understanding the Nexus and agent-Nexus connections for .NET agents on page 26.

If you are using the default Nexus on the embedded Agent Manager, or if you have created a single remote Nexus, a default Nexus Connection is created and pre-selected for you. Otherwise, select or define at least one Nexus connection.

To define a Nexus connection:

a Click **New Nexus connection**. The New Nexus Connection dialog box opens.

b Click **Add Nexus location** and select a Nexus from the list.

or

Click **Add Nexus location** and then click **Add custom** to define a new Nexus location by specifying the listen address and port. The default listen port is 41705. For example: `hostname.company.com:41705`.

c Optional — To add another Nexus location, click **Add Nexus location** again.

> **IMPORTANT:** If you add more than one Nexus, the agent attempts to connect to a Nexus in the order specified on this screen until it succeeds.

d Type a unique name for the Nexus connection in the **Save as** box.

e Optional — Click the **Advanced** tab to review and edit the connection settings. Click the icon to view a detailed description of the setting and its effects.

f Click **Save**.

g Click **Next**.

6 On the Transactions Agent — Configuration page, select the configuration that defines the instrumentation settings for the agent.

The **.NET Default** configuration is selected automatically if no other configurations exist.

> **TIP:** You can modify configurations, or assign new configurations to an existing agent, at any time after the agent has been created. For more information, see Managing .NET Agent Configurations on page 18.

Click **Next**.

7 On the Domain Assignment page, leave the **Assign servers the following domain** box blank to automatically assign monitored hosts to their Windows domain or workgroups.

To select a different domain:

- Click the down arrow in the **Assign servers the following domain** box to open a list of known domains.
- Type a domain in the **Assign servers the following domain** box.

Click **Next**.

8 On the Installation Directory page, define the location for the directory that contains all installation components. To accept the default directory, leave the **Directory name** box blank.

> **IMPORTANT:** If you have previously configured the host for a .NET agent installation, the existing installation directory is used.

To specify a custom directory, type a path in the **Directory name** box.

Click **Next**.

9 On the Review page, verify the configuration for the agents.

a Type a name for the integration configuration in the **Name** box. Use a specific, unique name so that you can identify this integration for reuse.

b By default, the **Activate agents*** check box is selected. It is recommended that you leave this option enabled. If you disable it, you must manually activate the agents and start data collection from the Application Servers Administration dashboard.

> **NOTE:** Restart the target IIS servers, Windows Services, or standalone applications to complete the integration process.

c If the information is correct, click **Finish**.

The Monitoring Task History section of the dashboard updates. For more information, see Viewing .NET agent task history on page 16.

The agent connects and begins collecting data. To view the data, open the Application Servers Monitor dashboard (**Dashboards > Application Servers > Monitor**). For more information about this dashboard, see the *Foglight for Application Servers User Guide*.

Configuring a remote host for .NET performance counters monitoring

The .NET performance counters monitoring agent can be installed locally (that is, on the same host that it is monitoring) or remotely. There are several configuration steps that may be required to ensure that a remote agent can connect and monitor a host.

Keep the following general comments in mind:

- Any access-denied errors are related to permissions and DCOM errors rather than WMI.
- RPC errors are related to firewall rules, rather than DCOM or WMI.
- Provider errors are related to WMI and could be caused by permissions on objects or namespaces.

To ensure successful remote monitoring of performance counters:

1. Start the remote registry service.
2. Configure the Windows® firewall in one of the following two ways:
   - Disable the firewall.

   or

   - Enable traffic through Windows Management Instrumentation (WMI) and allow a remote administration exception.

     For more information about WMI, see:
     http://msdn.microsoft.com/en-ca/library/windows/desktop/aa822854%28v=vs.85%29.aspx

     For more information about allowing remote admin exceptions, see:
     http://msdn.microsoft.com/en-us/library/aa389286.aspx
3. Review the information about Handling Remote Connections Under UAC:
   http://msdn.microsoft.com/en-ca/library/windows/desktop/aa826699%28v=vs.85%29.aspx

   Follow the recommendations for providing:

   a. Remote launch and activate rights to access DCOM.

   b. Rights to access the WMI namespace remotely (Remote Enable).
4. For a host in a workgroup, disabling Remote UAC by changing the registry entry that controls Remote UAC may be required. The registry entry is:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\system\L
   ocalAccountTokenFilterPolicy
   ```

   Set the value of this entry to 1 to disable Remote UAC.

Creating a .NET performance-counters-only monitoring agent

If you are creating a performance counters monitoring agent, and you want to be able to add transaction monitoring on the same host in future, ensure that you create the agent remotely. That way, you can create a transaction and performance counters monitoring agent locally at a later time.

Before you begin

Make sure that you have reviewed the information in Configuring a remote host for .NET performance counters monitoring on page 11 and that the necessary configuration is complete.

This procedure continues from the end of the procedure To create a .NET agent: on page 8.

To create a Performance Counters Monitoring Agent:

1 Select **Performance counters only** and click **Next**.

2 On the Monitored Hosts page, select the hosts that you want to monitor in one of the following ways:

- Click **Select hosts from a list** to open a list of hosts that are known to Foglight.

  In the Select Known Hosts dialog box, select one or more hosts from the list, or use the search box to locate a host. Click **OK** to close the box and save your entries.

- Click **Enter host names** to type the host names in a list.

  In the box that opens, type one or more host names separated by either commas or new lines. Click **OK** to close the box and save your entries.

> **IMPORTANT:** In order for a .NET agent to connect to and monitor a Windows host, it must have access to an account with Windows® Administrator privileges. Ensure that a credential exists and has been released to the Agent Manager. For information about creating and managing credentials, see the topic Controlling System Access with Credentials in the *Foglight Administration and Configuration Help*.

Click **Next**.

3 On the Agent Host page, select the Agent Manager that you want to use to manage the .NET agent you are creating.

Click **Next**.

4 On the Agent Names page, accept the default naming scheme, or customize the name on a per-agent basis.

To change the agent naming scheme, click **Edit** in the **Agent Name** column of the monitored host's row.

Click **Next**.

5 On the Integration Configuration page, select whether you want to create a configuration or reuse an existing one. This configuration controls whether the IIS is integrated, and specifies which Windows Services or standalone applications are integrated.

To learn about reusing an existing configuration, see Reusing a .NET performance-counters-only integration configuration on page 15.

For this example, select **Create new configuration**, and click **Next**.

6 On the New Integration Configuration page, specify integration targets for the agent. These targets can include IIS, Windows® Services, and standalone applications.

- **Integrate with IIS** is enabled by default. If you do not want to integrate with IIS, clear this check box.

- To integrate with a Windows Service, click **Add Windows Service**.

  A row appears in the list, with the Process Type specified as Windows Service.

  Click the **Name** box, and type the name of the Windows Service. This must be the service name, not the display name.

- To integrate with a standalone application, click **Add standalone application**.

  A row appears in the list, with the Process Type specified as Standalone Application.

  Click the **Name** box, and type the full path to the executable.

After you have specified all the targets for integration, click **Next**.

7 On the Domain Assignment page, leave the **Assign servers the following domain** box blank to automatically assign monitored hosts to their Windows domain or workgroups.

To select a different domain:

- Click the down arrow in the **Assign servers the following domain** box to open a list of known domains.
- Type a name in the **Assign servers the following domain** box.

Click **Next**.

8 On the Installation Directory page, accept the default directory for installed components and log files, or specify a custom directory.

> **IMPORTANT:** Only one installation directory is allowed per host. If an installation exists, a message appears to indicate this.

To specify a custom directory, type the path in the **Directory name** box.

Click **Next**.

9 On the Review page, verify the configuration for the agents.

a Type a name for the integration configuration in the **Name** box. Use a specific, unique name so that you can identify this integration for reuse.

b By default, the **Activate agents and start data collection*** check box is selected. It is recommended that you leave this option enabled. If you disable it, manually activate the agents and start data collection from the Application Servers Administration dashboard.

c If the information is correct, click **Finish**.

The Monitoring Task History section of the dashboard updates. For more information, see Viewing .NET agent task history on page 16.

The agent connects and begins collecting data. To view the data, open the Application Servers Monitor dashboard (**Dashboards > Application Servers > Monitor**). For more information about this dashboard, see the *Foglight for Application Servers User Guide*.

Related topics

- Editing .NET thread metrics integration configurations on page 25

Reusing .NET agent integration configurations

After you have created your first .NET agent, you can reuse the integration configuration to create additional agents that use the same integration targets and settings.

Related topics

- Reusing .NET transactions and performance counters integration on page 13
- Reusing a .NET performance-counters-only integration configuration on page 15.

Reusing .NET transactions and performance counters integration

This procedure continues from the end of the procedure To create a .NET agent: on page 8.

To reuse an existing integration configuration:

1 On the Setup Options page, select **Transactions and performance counters** and click **Next**.

2 On the Monitored Hosts page, select one or more hosts for the .NET agent.

> **IMPORTANT:** By default, only eligible hosts (hosts that have an active, compatible Agent Manager and that are not being monitored by a performance counters agent) are shown. To view all hosts, click Show eligible hosts only. The Monitored Hosts page refreshes and all hosts are shown.

Select hosts from the list and click **Next**.

3 On the Integration Configuration page, select whether you want to create a configuration or reuse an existing one. This configuration controls whether the IIS is integrated, and specifies which Windows® Services and standalone applications are integrated.

> **TIP:** To learn about creating a configuration, see Creating a .NET transaction and performance counters monitoring agent on page 8 or Creating a .NET performance-counters-only monitoring agent on page 11.

For this example, select **Reuse existing configuration**, and click **Next**.

4 On the Transaction Agent — Nexus Connection page, select or create the connection that the agent should use to connect to the Nexus.

> **TIP:** For more information about the Nexus, see Understanding the Nexus and agent-Nexus connections for .NET agents on page 26.

Select or define at least one Nexus connection for the agent you are creating.

Click **Next**.

5 On the Transactions Agent — Configuration page, select the configuration that defines the instrumentation settings for the agent.

The **.NET Default** configuration is selected automatically if no other configurations exist.

> **TIP:** You can modify configurations, or assign new configurations to an existing agent, at any time after the agent has been created. For more information, see Managing .NET Agent Configurations on page 18.

Click **Next**.

6 On the Domain Assignment page, leave the **Assign servers the following domain** box blank to automatically assign monitored hosts to their Windows domain or workgroups.

To select a different domain:

- Click the down arrow in the **Assign servers the following domain** box to open a list of known domains.

or

- Type a domain in the **Assign servers the following domain** box.

Click **Next**.

7 On the Installation Directory page, define the location for the directory that contains all installation components. To accept the default directory, leave the **Directory name** box blank.

> **IMPORTANT:** If you have previously configured the host for a .NET agent installation, the existing installation directory is used.

To specify a custom directory, type a path in the **Directory name** box.

Click **Next**.

8 On the Review page, verify the configuration for the agents.

d By default, the **Activate agents*** check box is selected. It is recommended that you leave this option enabled. If you disable it, manually activate the agents and start data collection from the Application Servers Administration dashboard.

> **NOTE:** Restart the target IIS servers, Windows Services, or standalone applications to complete the integration process

e If the information is correct, click **Finish**.

The Monitoring Task History section of the dashboard updates. For more information, see Viewing .NET agent task history on page 16.

The agent connects and begins collecting data. To view the data, open the Application Servers Monitor dashboard (**Dashboards > Application Servers > Monitor**). For more information about this dashboard, see the *Foglight for Application Servers User Guide*.

Reusing a .NET performance-counters-only integration configuration

Prerequisite:

- Review the information in Configuring a remote host for .NET performance counters monitoring on page 11, and complete the necessary configuration.

This procedure continues from the end of the procedure To create a .NET agent: on page 8.

To reuse an existing Performance Counters Monitoring Agent integration configuration:

1 Select **Performance counters only** and click **Next**.

2 On the Monitored Hosts page, select the hosts that you want to monitor in one of the following ways:

- Click **Select hosts from a list** to open a list of hosts that are known to Foglight.

  In the Select Known Hosts dialog box, select one or more hosts from the list, or use the search box to locate a host. Click **OK** to close the box and save your entries.

or

- Click **Enter host names** to type the host names in a list.

  In the box that opens, type one or more host names separated by either commas or new lines. Click **OK** to close the box and save your entries.

> **IMPORTANT:** In order for a .NET agent to connect to and monitor a Windows host, it must have access to an account with Windows® Administrator privileges. Ensure that a credential exists and has been released to the Agent Manager. For information about creating and managing credentials, see the topic Controlling System Access with Credentials in the *Foglight Administration and Configuration Help*.

Click **Next**.

3 On the Agent Host page, select the Agent Manager that you want to use to manage the .NET agent you are creating.

Click **Next**.

4 On the Agent Names page, accept the default naming scheme, or customize the name on a per-agent basis.

To change the agent naming scheme, click **Edit** in the **Agent Name** column of the monitored host's row.

Click **Next**.

5 On the Integration Configuration page, select whether you want to create a configuration or reuse an existing one. This configuration controls whether the IIS is integrated, and specifies which Windows Services and standalone applications are integrated.

a Select **Reuse configuration**.

b Select an existing configuration from the list.

c Click **Next**.

6 On the Domain Assignment page, leave the **Assign servers the following domain** box blank to automatically assign monitored hosts to their Windows domain or workgroups.

To select a different domain:

- Click the down arrow in the **Assign servers the following domain** box to open a list of known domains.

or

- Type a name in the **Assign servers the following domain** box.

Click **Next**.

7 On the Installation Directory page, accept the default directory for installed components and log files, or specify a custom directory.

> **IMPORTANT:** Only one installation directory is allowed per host. If an installation exists, a message appears to indicate this.

To specify a custom directory, type the path in the **Directory name** box.

Click **Next**.

8 On the Review page, verify the configuration for the agents.

d By default, the **Activate agents and start data collection** check box is selected. It is recommended that you leave this option enabled. If you disable it, you must manually activate the agents and start data collection from the Application Servers Administration dashboard.

e If the information is correct, click **Finish**.

The Monitoring Task History section of the dashboard updates. For more information, see Viewing .NET agent task history on page 16.

The agent connects and begins collecting data. To view the data, open the Application Servers Monitor dashboard (**Dashboards > Application Servers > Monitor**). For more information about this dashboard, see the *Foglight for Application Servers User Guide*.

Viewing .NET agent task history

The Task History section of the Application Servers Administration dashboard displays a list of agent-related tasks, such as *Setup .NET Performance Counter Monitoring (1 agent)*.

The results of the task appear in the Results column.

To view detailed task information:

- Click the **Result** text (for example, *Success*, or *Timeout*) to open the Task Result log.

Upgrading .NET agents

The *Foglight Upgrade Guide* provides full instructions for upgrading to the latest version of the cartridge and agents. Read Upgrade Foglight for Microsoft .NET in the Upgrading the Application Monitoring Cartridges section for details.

Depending on the version you are upgrading from, you may also need to upgrade the Nexus. See the *Foglight Upgrade Guide* for details.

Configuration migration during upgrades

If you have modified the configuration files and you upgrade to a minor version (for example from x.y.z to x.y.z+1), your modified files are preserved. The default contents of the configuration files from the new version are overwritten with your changes.

For example, you have version 5.9.3.2 currently installed and you have modified the Nexus *recording.config* and the .NET agent *instrumentation.config*. Upgrading to version 5.9.4 would overwrite the default contents of the 5.9.4 versions of those files with your file contents from 5.9.3.2. All other 5.9.4 configuration files are unchanged, and contain the default contents installed by the 5.9.4 cartridge.

NOTE: Configuration files managed through the Advanced dashboards (for example, *nexus/aggregation.config*) are not automatically migrated. If you have changed these files, they must be migrated manually, with the assistance of Customer Support where necessary.

In addition, the *nexus/compatible-builds.config* file is never automatically migrated. If you have any patch build IDs, these must be manually added to the new version's *compatible-builds.config* with the assistance of Customer Support.

IMPORTANT: No automatic migration occurs during an upgrade to the next major version (for example, from 5.9.x to 5.10.x).

2

Managing .NET Agent Configurations

The .NET Administration dashboard provides a centralized location for managing your agent installations and configurations, and collection schedules.

To access the .NET Administration dashboard:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.

   From here you can manage .NET agent installations, integrations, transaction agent configurations, and Nexus connections.

Managing .NET installations

The Installations tab of the .NET Administration dashboard provides an overview of installations, including their versions, collection configurations, and Nexus connections.

Use the Installations tab for the following activities:

- Rerunning the installer to fix any issues
- Uninstalling .NET installations
- Viewing .NET installation properties

Uninstalling .NET installations

Uninstalling an installation removes all files from the installation directory and deletes all the installed agents.

IMPORTANT: Restart IIS and any monitored processes to fully remove .NET transaction agents.

To uninstall a .NET installation:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.
3. On the .NET view, click the **Installations** tab.
4. On the Installations tab, click the installation host name in the **Name** column.
5. In the list that opens, click **Uninstall**.
6. Read the caution and confirm that you want to remove the entire installation by clicking **Uninstall**.

   A progress box opens.
7. Click **Close** to close the box when the process is finished.

Viewing .NET installation properties

Review the installation properties to quickly verify the installation directory, and the agent and Agent Manager versions.

To view the installation properties:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.
3. On the .NET view, click the **Installations** tab.
4. On the Installations tab, click the installation host name in the **Name** column.
5. In the list that opens, click **Properties**.

   The Installation Properties dialog box opens.
6. Verify the location of the installation directory, and the agent and Agent Manager version information.
7. Click **OK** to close the dialog box.

Managing .NET transaction agent configurations

The Transaction Agent Configurations tab of the .NET tab provides an overview of the existing transaction configurations and indicates how many agents are using each one.

Use the Transaction Agent Configurations tab for the following activities:

- Creating .NET transaction agent configurations
  - Checking which agents are using each configuration
  - Copying and deleting a configuration
  - Changing the configuration that is designated as the default configuration
- Viewing and editing .NET transaction agent configurations
- Filtering application pools monitoring

Creating .NET transaction agent configurations

The transaction agent configuration controls how the .NET agent collects transaction data from the monitored host. Each configuration can define a distinct collection of instrumentation settings. You can create configurations at any time and assign them to agents only when you want them to be used.

> **TIP:** For information about instrumentation settings, see Viewing and editing .NET transaction agent configurations on page 20.

To create a configuration:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.
3. On the .NET view, click the **Transaction Agent Configurations** tab.
4. On the Transaction Agent Configurations tab, click **New configuration...**

   The New .NET Transaction Configuration dialog box opens.
5. Optional — To use an existing configuration as the basis for your new configuration:
   a. Select the **Copy settings from** check box.

b Click the down arrow to the right of **.NetTransactions** and select an existing configuration from the list.

6 Type the identifier (ID) for this configuration in the first box. The ID can contain the following types of characters: alphanumeric, hyphens, periods, and underscores. No other characters are allowed.

7 Optional — Type a descriptive name for the configuration in the Display Name box.

> **NOTE:** If you leave this box empty, the ID also becomes the display name.

8 Optional — Type a short description for the configuration in the Description box.

9 Click **Create**.

The Transaction Agent Configuration view refreshes and the new configuration display in the list.

> **TIP:** Any new configuration that was not copied from an existing configuration contains the default monitoring settings.

Next steps:

- To view and edit the instrumentation settings, see Viewing and editing .NET transaction agent configurations on page 20.
- To set the new configuration as the default configuration: click the name of the configuration and, in the list that opens, select **Set as default configuration**.

  > **IMPORTANT:** You cannot delete the default configuration.

- To check which agents are using a configuration: click the name of the configuration and, in the list that opens, select **Used by**.

Viewing and editing .NET transaction agent configurations

You can view and edit the default transaction agent configuration, or any custom configurations you have created, through the .NET Transaction Agent Configuration view. Use the following procedure to edit the *instrumentation.config* settings for the .NET agents.

To view and edit a transaction agent configuration:

1 On the navigation panel, under Dashboards, click **Application Servers > Administration**.

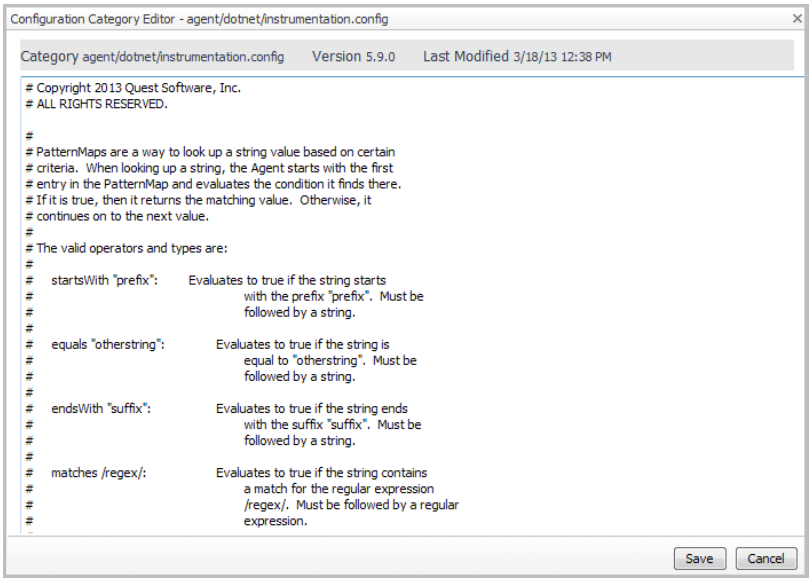
2 On the Application Servers Administration dashboard, click the **.NET** tab.

3 On the .NET view, click the **Transaction Agent Configurations** tab.

4 On the Transaction Agent Configurations view, click the name of the configuration that you want to view or edit.

5 In the list that opens, click **Edit Instrumentation Settings...**

The Configuration Category Editor opens.



Read the instructions and examples provided.

6 Type the instrumentation settings that you want to use.

7 Click **Save**.

8 Restart the affected IIS or Windows services and applications to apply the changes.

Filtering application pools monitoring

By default, the .NET Transaction Agent filters out all SharePoint application pools. All other application pools are instrumented.

You can add application pool names to the list of exclusions to filter out certain pools.

To view and edit application pool filtering:

1 On the navigation panel, under Dashboards, click **Application Servers > Administration**.

2 On the Application Servers Administration dashboard, click the **.NET** tab.

3 On the .NET view, click the **Transaction Agent Configuration** tab.

4 On the Transaction Agent Configurations view, click the name of the configuration that you want to work with.

5 In the list that opens, click **Manage**.

The Configuration Settings by Category view opens, displaying a list of the available configuration files.

6 Select **iis** and click **Edit**. Read the information in the configuration file to determine the syntax.

> **NOTE:** This setting applies only to the .NET Transaction Agent. The .NET Counters Agent continues to monitor the health of the application pools, however, no requests or traces are captured by the .NET Transaction Agent.

7 Save any changes you make.

8 Restart your IIS and any processes for the application pools you are excluding.

Managing .NET agent integration configurations

The Integration Configurations tab of the .NET view provides an overview of .NET agent configurations, including their target hosts and their target applications, and which Nexus connection they are using (if applicable).

Use the Integration Configurations tab for the following management activities for .NET agent integration settings:

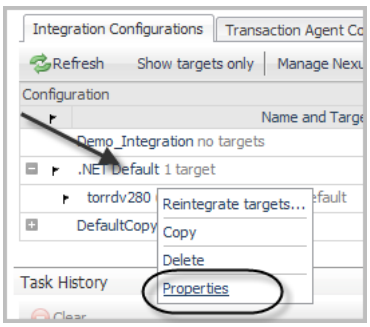
- Viewing and editing .NET integration configurations
- Editing IIS .NET integration configurations
- Editing standalone applications .NET integration configurations
- Editing Windows Services .NET integration configurations
- Editing .NET thread metrics integration configurations
- Managing .NET agent Nexus connections
- Managing collection schedules
- Increasing WMI polling memory values
- Preventing WMI quota violations

Viewing and editing .NET integration configurations

You can view and edit an integration from the Integration Configurations tab of the .NET view. These configurations control the monitoring settings for the .NET agent.

To view and edit an integration configuration:

1 Click the name of the configuration (not the name of the target). For example, click **.NET Default**.



2 In the list that opens, click **Properties**.

The Integration Configuration Properties dialog box opens, with the IIS tab displayed.

3 Click the tab for the type of target you want to work with. For details of how to use each tab, see the following sections:

- Editing IIS .NET integration configurations on page 23
- Editing Windows Services .NET integration configurations on page 24
- Editing standalone applications .NET integration configurations on page 24

- Editing .NET thread metrics integration configurations on page 25

Editing IIS .NET integration configurations

The .NET agent monitors for exceptions and performance issues in ASP.NET applications, application pools, and sites running under IIS. By default, the IIS performance counters and application availability metrics are monitored when you create a performance counters agent.

> **NOTE:** Using the default values for `the WMI Polling Memory Per User` and `WMI Polling Memory Total`, the agent can monitor up to 250 objects total (where objects include sites, web applications, and application pools).
>
> If your environment has a higher object count, see Increasing WMI polling memory values.

If you want to monitor specific objects, you can include or exclude them by name. Use the following procedure to limit the number of monitored web applications, applications pools, or sites. These settings are useful if you have encountered WMI Quota Violations, as described in Preventing WMI quota violations on page 30.

> **NOTE:** The following procedure assumes that you have an integration configuration open for editing, as described in Viewing and editing .NET integration configurations on page 22.

To monitor specific objects:

1. In the Integration Configuration Properties dialog box, ensure that the **IIS** tab is selected.
2. Confirm that the **Integrate with IIS** check box is selected.

   > **NOTE:** Clearing this check box disables all IIS and web application monitoring.

3. Click the down arrow to the left of **Advanced — Performance Counter Collection**. These settings affect objects that the Counters agent collect. They are only applicable if the IIS has been integrated.
4. Select the tab that corresponds to the type of object that you want to specify settings for: Web Applications, Application Pools, or Sites. The options and procedures for these objects are the same, as described below.
5. On the selected tab, use the drop-down menu to select **Specify names to collect**.

TIP: If you select **Do not collect**, only IIS is monitored.

6. Click **Add**. A blank row appears in the table.
7. Click in each box to toggle or edit the option.

   Optional — If you want to create a filter but not use it, click the empty box in the **Ignore** column. The text `IGNORE` appears, indicating that the row is being ignored.

   a. Click `INCLUDE` or `EXCLUDE` to toggle between the two options. The default setting is **include**.
   b. Select the **Regex** box to define the application name using a regular expression.
   c. Click in the **Value** box to the right of Regex to type the web application name.

8. To add another row, click **Add**, and repeat Step 7.
9. Select how objects that are not explicitly included or excluded in the table should be treated using the **Included/Excluded** option below the table.
10. When you are finished, click **OK** to save the configuration. A confirmation message appears.
11. Click **OK**.

Editing Windows Services .NET integration configurations

In the Integration Configuration Properties dialog box, define the service name of the Windows® Services that you want to monitor. Integrating with a Windows Service provides availability and transaction metrics, and performance counters. When you integrate a .NET transaction agent with a Windows Service, the agent changes the Windows registry.

> **NOTE:** The following procedure assumes that you have an integration configuration open for editing, as described in Viewing and editing .NET integration configurations on page 22.

To monitor Windows Services:

1. In the Integration Configuration Properties dialog box, click the **Windows Services** tab.
2. To add a single Windows Service, click **Add**.

   A blank row appears in the table.
3. Click in the **Windows Service where service 'name' is** box and type the full name of the service. This is the service name, not the display name.
4. To add another service, repeat steps 2 through 4.

   Or

   To add multiple services in a single step, click **Add multiple**. In the dialog box that opens, type the names of the Windows Services, separated by commas or new lines, then click **Set**.
5. Click **OK** to close the editor and save your changes.
6. Restart your Windows Services to apply the changes.

Editing standalone applications .NET integration configurations

In the Integration Configuration Properties dialog box, define the full path to the executable file of any standalone applications that you want to monitor. Integrating with a standalone application provides availability and transaction metrics, and performance counters. When you integrate a .NET transaction agent with a standalone application, a `cmd` script is created and written to a file in the application's `root` directory. Use the `cmd` script to start your standalone applications to get the applications integrated .

> **NOTE:** The following procedure assumes that you have an integration configuration open for editing, as described in Viewing and editing .NET integration configurations on page 22.

To monitor specific standalone applications:

1. In the Integration Configuration Properties dialog box, click the **Standalone Applications** tab.
2. To add a single standalone application, click **Add**.

   A blank row appears in the table.
3. Click in the **Standalone application where 'executable path' is** box and type the full path to the executable.
4. To add another application, repeat Step 2 through Step 3.

   Or

   To add multiple applications in a single step, click **Add multiple**. In the dialog box that opens, type the names of the standalone applications, separated by commas or new lines, then click **Set**.

5 Click **OK** to close the editor and save your changes.

> **NOTE:** A `cmd` script is created and written to a file in the application's `root` directory.

6 To integrate your applications, restart your standalone applications using the `cmd` script.

Editing .NET thread metrics integration configurations

In the Integration Configuration Properties dialog box, specify whether to enable or disable collection of thread metrics from integrated IIS, Windows® Services, and standalone applications. By default, thread metrics are not collected.

This setting applies to the Performance Counters Agent only. For more information about this agent, see Creating a .NET performance-counters-only monitoring agent on page 11.

> **NOTE:** The following procedure assumes that you have an integration configuration open for editing, as described in Viewing and editing .NET integration configurations on page 22.

To collect thread metrics:

1 In the Integration Configuration Properties dialog box, click the Thread Metrics tab.

2 Select the Collect Thread Metrics check box.

3 Click **OK** to close the editor and save your changes.

4 Restart your IIS, Windows Services, or standalone applications.

Switching .NET agent configurations

You can create or edit agent configurations through the .NET Administration dashboard before or after creating a .NET agent. If you create an agent and use the default configuration during installation, you can later create another customized configuration and assign it to the agent.

See Creating .NET transaction agent configurations on page 19 or Managing .NET agent integration configurations on page 22 for information about creating configurations.

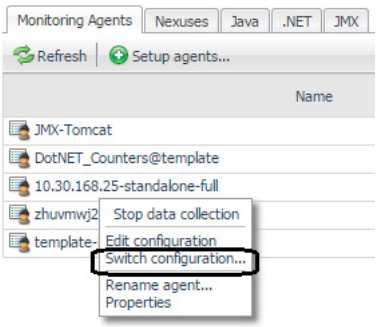
This procedure describes how to switch the configuration that an agent is using to collect data after the agent and a unique configuration have been created.

To switch an agent's configuration:

1 On the navigation panel, under Dashboards, click **Application Servers > Administration**.

2 On the Application Servers Administration dashboard, in the Monitoring Agents table, click the **name** of the agent to which you want to assign a new configuration.

3 In the list that opens, click **Switch configuration**.



Depending on the type of agent you selected (transaction or counters), either the Switch Transaction Configuration or the Switch Counters Configuration dialog box opens.

> **TIP:** These two dialog boxes function identically. The only difference is the name, which reflects the type of agent selected.

4 Select a configuration from the list and click **Switch**.

A message box opens, indicating that the configuration that the agent uses is updated. You do not need to restart the agent.

5 Click **Close**.

Understanding the Nexus and agent-Nexus connections for .NET agents

Foglight for Microsoft .NET includes the Nexus, a central analytical engine that correlates data sent from multiple agents, restores and preserves the temporal order of events across all servers, and provides other basic analysis of the data. The Nexus is essential to the data management facilities of Foglight for Microsoft .NET.

Foglight creates a default Nexus as part of the installation process for Foglight for Microsoft .NET. The default Nexus controls all agent data correlation and submission unless you create another Nexus and assign agents to it instead.

> **TIP:** For more information about the Nexus, see the topic Understanding the Nexus in the *Foglight for Application Servers Administration and Configuration Guide*.

The Nexus connection is the communication link between the .NET agent and the Nexus. It defines how the agent communicates with the Nexus and controls settings such as retry intervals and timeouts. The agent-Nexus connection is created as part of the cartridge installation process. You can also create connections, or edit existing ones, and change which agents are using a connection by switching the agent's Nexus connection.

Managing .NET agent Nexus connections

During the process of creating a .NET agent, you assign it to an existing Nexus, and an agent-Nexus connection is created. You can also change the connection between an agent and a Nexus, or create a connection and assign it to different agents after they have been created.

> **NOTE:** The Nexus is only applicable to .NET agents that are monitoring transactions.

Creating or Editing Agent-Nexus Connection Settings

The process for creating an agent-Nexus connection and the process for editing an existing one are similar. The following procedure describes both options and any differences between them.

To change agent Nexus connection settings:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, clickthe **.NET** tab.
3. On the .NET view, click the **Integration Configurations** tab.
4. On the Integration Configurations tab, click **Manage Nexus connections**.

   The Manage Nexus Connections page opens.

   This page lists all existing agent-Nexus connections and the location of the Nexus. From here, you can edit an existing connection, or create a new one.
5. To edit an existing connection, click the name of the connection, then click **Edit**.

   Or

   To create a connection, click **New**.

   The Nexus Connection dialog box opens.
6. On the Locations tab, you can add Nexuses that the agent should attempt to connect to, in the order that you want. The agent attempts to connect to each Nexus in turn until a successful connection is established.

   Click **Add Nexus location** and select a Nexus from the list.

   > **TIP:** Use the **Add custom** option to define a Nexus location by listen address and port.
7. Optional — Use the up and down arrows to change the order in which the agent attempts the Nexus connections.
8. Click the **Advanced** tab.
9. Edit the connection settings as necessary. Click the info icon to view a detailed description of the setting and its effects.
10. New Nexus connections only — If you want to reuse this connection and assign it to other agents, type a unique name in the **Save as** box.
11. Click **Save**.

    The Manage Nexus Connections page refreshes. Any changes you have made to the Nexus location are listed in the Nexus Locations column. The time and date in the Last Modified column are also updated.

Updating the Nexus connection for a target application server

If you changed the Nexus Connection (see Managing .NET agent Nexus connections on page 26 for details), you can update any target application servers that are using that Nexus connection through the **Integration Configurations** tab of the .NET Administration dashboard.

To update an application server's Nexus connection:

1. On the navigation panel, under Dashboards, click **Application Servers> Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.
3. On the .NET view, click the **Integration Configurations** tab.

4 In the **Name and Targets** list, expand the configuration group that contains the target by clicking the '+' sign.

5 Click the name of the target application server that you want to update.

6 In the menu that opens, click **Update Nexus connection**.

Switching a Nexus connection

You can change the Nexus connection between an agent and a Nexus after it has been created. For example, you might switch the connection for performance reasons (divide up the agents) or to organize your environment (all agents monitoring a particular application use the same Nexus). Switching the Nexus connection affects all of the agents on that the target (host).

To switch a Nexus connection:

1 On the navigation panel, under Dashboards, click **Application Servers > Administration**.

2 On the Application Servers Administration dashboard, click the **Java** tab.

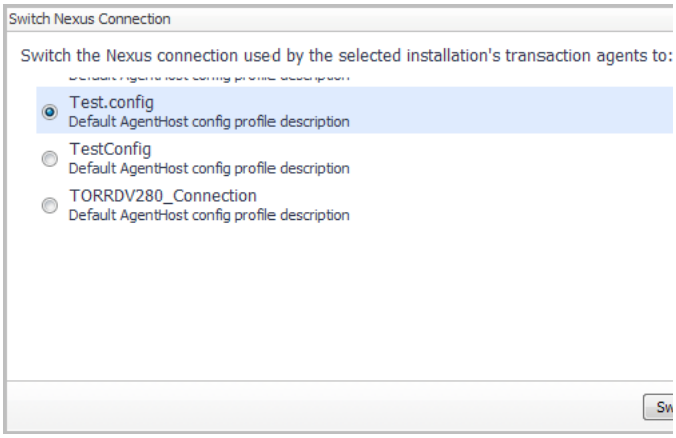
3 On the Java view, click the **Integration Configurations** tab.

4 In the **Configuration Name and Targets** list, expand the configuration group by clicking the '+' sign.

5 Click the name of the target (host) that you want to switch the Nexus connection on.

6 In the list that opens, select **Switch Nexus Connection**.

A dialog box opens, listing all the available Nexus Connections for that host integration.



7 Select the Nexus connection towhich you want to switch.

8 Click **Switch** to save your changes.

All of the agents on that host now use this Nexus connection to communicate with the Nexus.

Managing collection schedules

Use the Manage Collection Schedules dashboard to create and manage the schedules that define the frequency at which the .NET performance counters only agents collect data.

Agents use the default schedule unless you create and assign a different collection schedule through this view.

For more information, see:

- Creating Collection Schedules on page 29
- Assigning custom collection schedules to an agent on page 29

Creating Collection Schedules

You can create a custom collection schedule to control the collection types and intervals for particular agents.

To create a custom schedule:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab.
3. On the .NET view, click the **Collection Schedules** tab.
4. On the Collection Schedules view, click **New collection schedule...**

   The New Collection Schedule dialog box opens.
5. For each collection type, select a collection interval. Click the interval value (for example, 24 hours) to open a list of values.

   If you need more information about a particular collection type, or if you want to double-check the default recommended value, click the icon for that type.
6. Type a name for the new custom schedule in the **Schedule name** box.
7. Click **Save**.

   The dialog box closes, and the Collection Schedules view refreshes. You can now assign the custom schedule to one or more agents. For more information, see Assigning custom collection schedules to an agent on page 29.

Assigning custom collection schedules to an agent

After Creating Collection Schedules, you can assign the newly created schedules to .NET agents.

To assign a custom schedule to an agent:

1. On the navigation panel, under Dashboards, click **Application Servers > Administration**.
2. On the Application Servers Administration dashboard, click the **.NET** tab..
3. On the .NET view, click the **Collection Schedules** tab.
4. On the Collection Schedules view, click the name of the schedule that you want to assign to one or more agents.
5. In the list that opens, click **Assign to agents**.

   The Assign Collection Schedule — <schedulename> dialog box opens.
6. Select one or more agents from the list.
7. Click **Assign**.

   The dialog box closes and a message box opens, verifying that the schedule was assigned to the selected agents.
8. Click **OK** to close the message box.

Increasing WMI polling memory values

By default, the .NET agent can monitor up to 250 total objects when the default values for Polling WMI Memory Per User and WMI Polling Memory Total are used. This total object count includes the number of web sites, the number of web applications, and the number of application pools.

For environments with higher object counts, you may need to increase both of the Polling Memory values to ensure that a quota violation error is not triggered.

> **NOTE:** These changes apply only to local agents. They have no effect on remotely hosted counter agents.

To increase the polling memory values:

1. On the Application Servers Administration dashboard, click the name of the monitoring agent for which you want to adjust the memory values.
2. In the menu that opens, click **Properties**.
3. At the bottom of the Agent Properties dialog box, click the down arrow beside **Advanced Settings**.
4. Adjust the values for either the Total or the Per User polling memory as necessary.

   WMI Polling Memory Per User is the maximum amount of memory that can be consumed when polling event queries that a particular user issues. Default value is 25MB.

   WMI Polling Memory Total is the total amount of memory that can be consumed when polling event queries that all users issue. Default value is 50MB.
5. Click **OK**.
6. Stop and restart the data collection.

Preventing WMI quota violations

Sometimes, due to the limitations on the length of query strings for web applications, application pools, and sites, a WMI Quota Violation may be encountered.

When this issue occurs, the following error message appears in the agent log:

```
The agent can only monitor a subset of all specified IIS configuration objects
due to WMI query string limit. Consider adjusting the Included/Excluded
settings for Web Applications, Application Pools, and Sites.
```

If you see this message, see the following section for information about adjusting the included/excluded settings: Editing IIS .NET integration configurations on page 23.

About Dell

Dell listens to customers and delivers worldwide innovative technology, business solutions, and services they trust and value. For more information, visit http://www.software.dell.com.

Contacting Dell

For sales or other inquiries, visit http://software.dell.com/company/contact-us.aspx or call 1-949-754-8000.

Technical support resources

Technical support is available to customers who have purchased Dell software with a valid maintenance contract and to customers who have trial versions. To access the Support Portal, go to https://support.software.dell.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. In addition, the Support Portal provides direct access to product support engineers through an online Service Request system.

The Support Portal enables you to:

- Create, update, and manage Service Requests (cases).
- View Knowledge Base articles.
- Obtain product notifications.
- Download software. For trial software, go to http://software.dell.com/trials.
- View how-to videos.
- Engage in community discussions.
- Chat with a support engineer.