Quest™

Foglight® for SQL Server 6.0.0

Managing SQL Server Database Systems

User and Reference Guide

© 2021 Quest Software Inc.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal.

Trademarks

Quest, the Quest logo, Foglight, and Where next meets now are trademarks and registered trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at www.quest.com/legal. 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. 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. 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. 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. All other marks and names mentioned herein may be trademarks of their respective companies.

Legend

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

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

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

Foglight for SQL Server User and Reference Guide
Updated - May 2021
Software Version - 6.0.0

Contents

Introduction to this Guide 6
Installation 6
Overview 7
Running the Upgrade Wizard 7
Migrating Agents 9
About Monitoring Extensions 9
Installing and Monitoring a Single SQL Server Instance 11
Discovering Multiple Instances to Monitor 12
Adding instances manually 14
Discovering instances by host name 14
Discovering instances by IP range 15
Importing Instances from a File 17
Configuring Multiple Instances for Monitoring using Silent Installation 17
Contents of the Input CSV File 18
Configuring the On Demand Data Port on the Agent Manager Concentrator 20

Using Foglight for SQL Server 22
Viewing the Databases Dashboard 22
Assigning Instances to Users or Groups 23
Selecting an Instance to Monitor 24
Foglight for SQL Server Toolbar and Views 28
Overview Dashboard 30
Advisories Dashboard 31
Monitoring Data Replication 35
Viewing the Replication Home Page 36
Monitoring SQL Performance 46
Resource Toolbar Options 46
Monitoring SQL Performance without SQL PI configured 50
Monitoring SQL Performance with SQL PI Configured 53
Reviewing Memory Usage 62
Viewing the Memory Summary 62
Monitoring Buffer Cache-related Data 65
Reviewing the Instance Activity 69
Viewing In-depth Data about the Instance 69
Tracking the Instance Data Flow 76
Reviewing Database Usage 98
Monitoring SQL Server Databases 98
Monitoring Database Details 101
Reviewing the Services 112
Reviewing the Support Service Status 113
Reviewing SQL Agent Jobs 114
Viewing SQL Agent Alerts 115
Monitoring SQL Server Transactions using the DTC Panel 116

Monitoring Full-text Indexes using the Full Text Search Panel . . . 117
Using the HADR Drilldown . . . 117
Monitoring the Log Shipping . . . 118
Monitoring Cluster Services . . . 119
Tracking the Status of the Mirroring Operation . . . 120
Reviewing the Always On Availability Groups . . . 124
Using the Logs Drilldown . . . 129
Reviewing the SQL Server Error Logs . . . 129
Reviewing the SQL Agent Error Logs . . . 129
Reviewing Configuration Settings . . . 130
Reviewing SQL Server Configuration . . . 130
Viewing User-defined Performance Counters and Collections . . . 131
Viewing User-Defined Performance Counters . . . 131
Viewing User-Defined Collections . . . 132

Monitoring Business Intelligence Services . . . 133
Permissions . . . 133
Monitoring Integration Services . . . 133
Configuring Integration Services Monitoring . . . 133
Overview dashboard . . . 134
Packages Dashboard . . . 135
Terminology . . . 135
Monitoring Reporting Services . . . 136
Configuring Reporting Services Monitoring . . . 136
Monitoring Reports using the Reporting Services Panel . . . 136
Monitoring Analysis Services . . . 137
Configuring Analysis Services Monitoring . . . 137
Overview dashboard . . . 137
Unprocessed Objects Dashboard . . . 138
Current Activity Dashboard . . . 138
Monitoring Analysis Service Performance . . . 138
Resource Toolbar Options . . . 138
Performance Tree . . . 139
Viewing Historical Metrics . . . 140
Viewing Change Tracking . . . 140
BI Administration Settings . . . 140
Defining Connection Details . . . 141
Alarms . . . 141

Administering Foglight for SQL Server . . . 142
Configuration Settings . . . 142
Reviewing SQL Server Configuration . . . 142
Managing Foglight for SQL Server Agent Settings . . . 144
Opening the Databases Administration Dashboard . . . 144
Reviewing the Administration Settings . . . 144
Defining Connection Details . . . 145
Customizing Alarms for Foglight for SQL Server Rules . . . 147
Defining Data Collection and Storage Options . . . 155

Defining Error Log Filtering . . . 156
Configuring Performance Counters . . . 157
Setting Options for Displaying Data in the Buffer Cache . . . 158
Setting Options for Displaying Data in the Plan Cache . . . 158
Setting Options for Displaying Data in the Locks Panel . . . 159
Defining Retention Policies . . . 160
Defining the Collection and Display of Top SQL Statements . . . 161
Defining the Collection of Database Indexes . . . 161
Configuring User-defined Collections . . . 162
Administering SQL Performance Investigator . . . 164
Configuring the On-demand Data Settings . . . 164
Configuring the Database to be Excluded . . . 164
Reviewing Foglight for SQL Server Alarms . . . 165
Alarms Displayed in the SQL Processes Panel . . . 166
Alarms Displayed in the SQL Memory Panel . . . 168
Alarms Displayed in the Background Processes Panel . . . 170
Alarms Displayed in the Database Details Panel . . . 173
Generating Reports . . . 178
Generating Reports for a Foglight for SQL Server Instance . . . 178
Studying the Various Reports . . . 179
Monitoring SQL Server instances on VMware servers . . . 180
Viewing Data Displayed on vmExplorer . . . 180

Reference . . . 184
SQL Performance Investigator Metrics . . . 184
Rules . . . 195
Collections and Metrics . . . 195

Glossary . . . 283
A . . . 283

About us . . . 309
Technical support resources . . . 309

1

Introduction to this Guide

Welcome to the *Foglight for SQL Server User and Reference Guide*. This guide provides configuration instructions, conceptual information and instructions on how to use the Foglight for SQL Server cartridge. It describes the dashboards included with the cartridge and how they are used for collecting monitoring data from the entire relational database management system. Also covered are the cartridge's interaction with and support of additional services and modules, such as replication and virtualization.

This guide is intended for users who wants to know more about monitoring SQL Server instances and the steps required for discovering and configuring the SQL Server agent. It is also meant for those users who want to learn about the methods used for configuring and applying user-defined settings.

Installation

Information regarding the installation of Foglight *for SQL Server*, its pre-requisites and permission can be found in the *Foglight for SQL Server Deployment Guide*. You can view this guide on Quest's website at http://support.quest.com/technical-documents/foglight-for-sql-server-cartridge/5.9.7.10/deployment-guide/.

Installing and Configuring Agents

Foglight *for SQL Server* monitors the SQL Server database activity by connecting to and querying the SQL Server database. The agents provided monitor the SQL Server database system. The dashboards included with the cartridge provide a visual representation of the status of the major components of the SQL Server agents. They allow you to determine any potential bottleneck in database performance.

Upgrading to the Current Version

Starting to work with a Foglight *for SQL Server*Foglight for SQL Server cartridge requires upgrading to the current version of both the cartridge and the Foglight Agent Manager that runs the cartridge.

This topic contains instructions for using the upgrade wizard.

IMPORTANT: Foglight for SQL Server does not support the upgrade method of placing the new version under the folder FGLHOME/upgrade/cartridge folder.

IMPORTANT: If upgrading to the current version of Foglight for SQL Server in a Federation architecture, the upgrade should be applied first to the stand-alone Management Servers (the Federated Children) and then to the central Management Server (the Federation Master).
If the Federation Master is upgraded first, this server displays incorrect information regarding the number of instances being monitored, as reported under the Status Summary section. All the same, this information is displayed correctly on the Federated Children.

Overview

If you upgrade Foglight for SQL Server, without also upgrading the database cartridge components, the discrepancy will be detected when you access the Databases dashboard. The upgrade wizard launches automatically.

Selecting the check box at the bottom left of the screen prevents this wizard from appearing when entering the Databases dashboard. Nevertheless, if several database cartridge components still require upgrade, the need to upgrade them is indicated in the dashboard. The indication is evident both in the caption *Upgrade required*, which is displayed in red to the right of the requested instance. It also appears in the button **Upgrade**, which appears only if upgrade is required.

NOTE: Even though instances whose components need to be upgraded appear in the Databases table, such instances cannot be accessed by clicking them, until they have been fully upgraded.

Running the Upgrade Wizard

To upgrade the requested instances:

1. Click **Upgrade**.

   The screen that appears now is the same screen that appears by default upon entering the Databases dashboard when one or more database cartridge components require upgrading. While the need for upgrade is indicated for both Foglight for Oracle and Foglight for SQL Server, the upgrade is carried out separately for each database cartridge type.

2. Click **Foglight for SQL Server**.

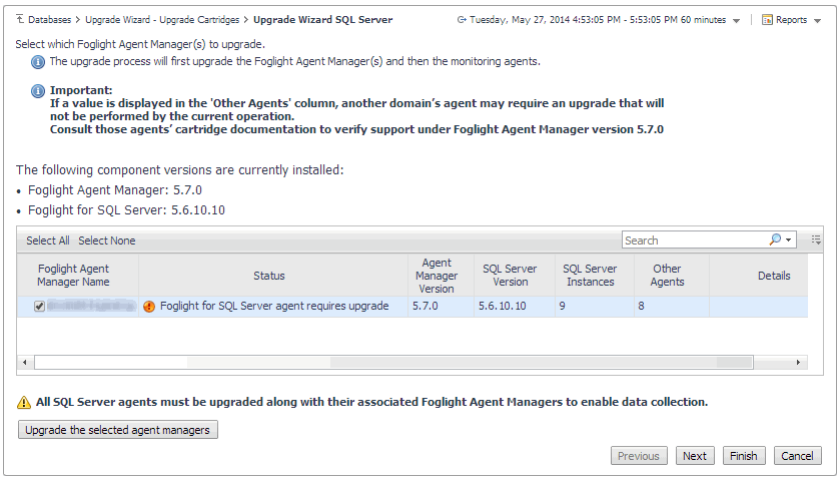
   A Welcome page displays What's New in this release, including bug fixes.

3. If SQL Performance Investigator is *not* configured, proceed to Step 5.

4. If SQL Performance Investigator is configured, the SQL PI repository must be upgraded prior to the agents upgrade.

5. Select the Agent Managers to upgrade.

Figure 1. Select the Agent Managers to install.



NOTE: The Other Agents column in the table indicates whether the specified Foglight Agent Manager runs agents of other cartridge types. If a value is displayed in this column, go to Dashboards > Administration > Agents > Agent Status to view which other agents run under the specified Foglight Agent Manager. If the other agents appear in the list below, upgrading Foglight Agent Manager to the latest version is safe. For agents of any other cartridge type, consult the documentation of the respective cartridge types.

- DB_DB2_*
- DB_SQL_Server_*
- DB_Oracle_*
- Sybase_MDA
- UnixAgent
- WindowsAgent

6 Click **Upgrade the selected Foglight Agent Managers**.

A progress bar appears.

7 After the upgrade is complete, click **Next**.

The next screen is used for granting privileges to users that were detected as possibly requiring additional privileges to ensure full functionality.

8 Select the agents displayed in the table and click **Validate connectivity**.

9 If the status row of one or more agents displays the status Insufficient privileges, complete the following steps:

a Select all agents that require privileges updates, and click **Grant privileges**.

b In the Grant Database Privileges dialog box, enter a SYSADMIN user name and password.

c To view the script used for granting the privileges, click **View script**.

IMPORTANT: The script for granting privileges is invoked using a pop-up.To view the script, ensure that pop ups are not blocked on the page.

d Click **Grant database privileges**.

10 On the Rule Modification Overview page, review the list of modified rules.

Modified rules have one of the following states:

- Removed — The rule is obsolete and was deleted from the Management Server.
- New — The rule is new in this release. To review its definition, click **View new rule**.
- Updated — The rule was updated in this release. To review the updated rule, click **View updated rule**. If a rule targeted for update is modified by a user, the user's modified rule is copied and disabled before the updated rule is installed. To review the modified rule, click **View user-updated rule**.

TIP: To avoid having to repeat rule modifications after an upgrade, do not edit rules with the DBSS prefix in the Rule Management dashboard. Use the Alarm Template dashboard instead. When you make your edits on the Alarm Template dashboard view, the edits are saved separately and applied over the predefined rules. For instructions, see Customizing Alarms for Foglight for SQL Server Rules on page 147.

11 Foglight Performance Analysis is no longer supported and has been replace by the SQL Performance Investigator. All Instances that are configured to use Foglight Performance Analysis will be deprecated and replace with the SQL Performance Investigator capabilities.

During this process you are prompted to select a dedicated Agent Manager on which to install an external database repository. The SQL PI configuration takes into account the resources needed for monitoring with SQL PI and prompts you with further information.

All information regarding the resource allocations needed for SQL PI monitoring can be found in the SQL Server cartridge *Hardware Sizing Guide*.

Migrating Agents

The SQL Performance Investigator extension is only supported on Windows 64-bit. As a result, you may need to migrate SQL Server agents from an Agent Manager running on Linux to a Windows Agent Manager.

In addition, the migration tool allows you to perform resource balancing between the agent managers running SQL Server agents with and without the SQL PI extension installed.

To migrate an agent to a different Agent Manager:

1. Open the migration wizard by navigating to **Dashboards > Databases > Support Dashboard > Databases Technical Support > Database Agents Migration > SQL Server**.
2. Select the Agent Manager that is the destination host for the migrated SQL Agents. Click **Next**.
3. On the SQL Agents for Migration pane, the left-hand column displays a list of available Agent Managers. Select an Agent Manager to view a list of the agents installed on it.
4. From the list, select the agents that you want to migrate and move them to the Agent Manager Destination column.
5. If necessary, select agents and set the Windows credentials.
6. Click **Next** to migrate the agents.
7. When migration has completed, click **Finish** to exit the Wizard.

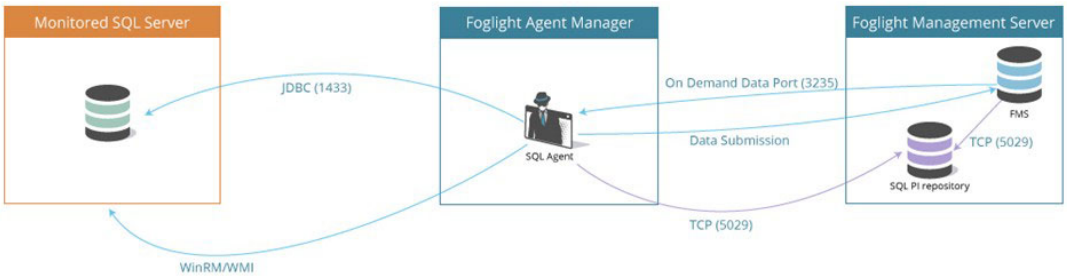
About Monitoring Extensions

During the installation process you can choose to install and configure one or more of the monitoring extensions. The monitoring extensions provide a more in-depth analysis of the monitored instance and the environment it is running on, creating a whole and unified status.

Understanding the Foglight for SQL Server Architecture

The communication architecture of Foglight for SQL Server is presented below.

Figure 2. Communication Architecture overview.



SQL Performance Investigator Extension

SQL Performance Investigator allows you to rapidly identify bottlenecks, anomalies, and application trends by focusing on top resource consumers and providing multi-dimensional SQL domain drilldowns. SQL PI allows you to:

- Monitor real-time SQL Server database performance at a glance
- Gather and diagnose historical views
- Identify and anticipate performance issues
- Analyze and optimize execution plan changes
- Compare day-to-day values to identify anomalies and application changes

NOTE: SQL Performance Investigator requires a license. If you are using a trial version and would like to request pricing, contact Quest Support.

Operating System Extension

Monitoring the operating system allows you to identify resource consumption and provides a full view of the server health. An Infrastructure agent, which is created automatically as part of the monitoring process, monitors the operating system.

NOTE: The Operating System extension is enabled by default.

VMware Extension

Monitoring the VMware system allows you to identify resource consumption and provides a full view of the data center and ESX health when the server is part of a VMware environment.

Installing and Monitoring a Single SQL Server Instance

Enabling the Foglight Management Server to monitor SQL Server instances requires the creation of the Foglight agents that monitor these instances and ensuring that these agents communicate properly with the Foglight Management Server.

Foglight for SQL Server provides a graphic, intuitive method for creating and configuring multiple agents, which can be used instead of Foglight's default method for creating agents and editing their properties using the Agent Administration dashboard (see About the Foglight for SQL Server Agent in the *Foglight for SQL Server Reference Guide*). Foglight for SQL Server allows running a wizard that provides a common entry point for adding and discovering all database instances within a user-specified range, and then configuring these instances for monitoring.

IMPORTANT: When running Foglight for SQL Server in a Federation architecture, neither the creation nor the administration of agents can be accomplished from the central Foglight Management Server (the Federation Master). These two tasks should be carried out from the stand-alone Management Servers (the Federated Children).

To run the instance installation wizard:

1 On the navigation panel, click **Homes > Databases**.

2 Click the **SQL Server** tab in the middle of the Databases View, and then click **Monitor**.

The Monitor SQL Server Instance dialog box appears.

NOTE: If a user-defined database group is selected, the databases table's title displays the name of this group instead of All; however, all newly discovered or created databases are added to the general (All) group of databases.

3 Choose the agent manager on which the agent is running. The default is the agent manager with the least agents installed.

a Click the **Agent Manager Host** link located in the bottom left corner of the dialog box.

A dialog box appears with a list of all agent managers connected to the Foglight management server.

b Select the appropriate host name and click **Set**.

IMPORTANT: You have the option set this host as the default for all future installations.

4 On the Monitor SQL Server Instance dialog box, provide Connection Details by typing:

- Server name — for the default instance, specify the host name. If you have more than one instance on the server, you must name the instances using the format: domain\user name.
- Port — **Optional**. This field can be left empty, unless the TCP/IP connection port is not the default port: 1443.

5 Specify the SQL Server credentials using either of the following authentication methods:

- Active Directory (AD) Authentication — log in using the Active Directory account running on your agent manager or typing a new AD account. The user name should be typed in the following format: domain\user name.
- SQL Server Authentication — log in using a SQL Server account
- Azure Active Directory - log in using an Azure Active Directory account.

6 Use SSL: Select whether an SSL connection should be used.

7 In the **Configuration** section, select the Alarm Template to use for this instance. The Factory Template will be applied by default. Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight User Guide* for more information.

8 **Optional** — In the Monitoring Extension pane, click the **Click for Licensing Information** link under **SQL PI**. You are prompted to contact Quest Sales for an add-on license to enable SQL Performance Investigator (PI).

9 **Optional** — In the Monitoring Extension pane, click the **Operating System** link.

To configure the extension, choose the connection details of the host on which the SQL Server instance is running:

- Log in to the host using the same account used for monitoring SQL Server — This option is set by default when the SQL Server connection details are of types: Active Directory (AD) Authentication or Using the Active Directory account running your agent manager.
- Log in to the host using different log in credentials — logging in through either of the following authentication methods:
  - Windows: Log in through a Windows account. The user name should be entered in the domain\username format (for example, COLUMBIA\JSmith).
  - Local User: Log in through the same credentials that are used to run monitored software on the SQL Server host. The user name should be entered in the domain\username format (for example, COLUMBIA\JSmith).
  - SSH (login credentials): Log in through a user account on Linux.
  - SSH (RSA): Log in through a RSA key on Linux.
  - SSH (DSA): Log in through a DSA key on Linux.

10 **Optional** — In the Monitoring Extensions pane, click **Collect VM statistics**.

To configure the extension, select the connection details of the vCenter or ESX on which the SQL Server instance is running:

- The name of IP address of the vCenter server that hosts the SQL Server instance virtual machine, or the name of its parent ESX server
- The port number used by the vCenter server system or by ESX server for listening to the connections from the vSphere Client (default: 443).
- The name and password of the user that has the privileges required for connecting to the vCenter server or ESX server and retrieving information.

11 Click **Monitor**.

> **IMPORTANT:** If the monitoring verification fails click the message that is displayed on the Status column and resolve the issue according to the instructions that appear in the dialog box. For example, (for example: insufficient privileges, incorrect credentials or an Agent Manager that reached its full monitoring capacity.

12 When the installation completes successfully, the Monitoring Initialized Successfully dialog box appears. Click **Add another Database** or **Finish** to exit.

Discovering Multiple Instances to Monitor

You can use the instance installation wizard to discover existing instances. This option allows you to discover instances and monitoring them by entering several methods:

- By host name

- By IP range
- By entering a file with a predefined instance.

To discover additional instances to monitor:

1 Click the **SQL Server** tab in the middle of the Databases View, and then click **Monitor**.

The Monitor SQL Server Instance dialog box appears.

2 Click the **Use this option to discover multiple SQL-Server instances** link. The Select an Agent Manager dialog box appears.

3 Choose an Agent Manager host, click **Validate** to validate the Agent Manager system resources, and then click **Next**.

> **IMPORTANT:** At the bottom of the pane you can select a default agent manager for all future installations.

4 Click **Add instances** and select one of the following methods for adding instances:

- (Windows and Linux) Adding instances manually — see Adding instances manually on page 14
- (Windows only) Discovering instances by hosts — see Discovering instances by host name on page 14
- (Windows and Linux) Discovering instances by IP range — see Discovering instances by IP range on page 15.
- (Windows and Linux) Importing instances names from a file — see Importing Instances from a File on page 17

5 After the discovery process is successfully completed, the newly discovered instances appear on the table, with the status *Set credentials.*

6 Click the check boxes beside the instances whose credentials are to be configured.

7 Click **Set credentials** to provide the instances' log in credentials and monitoring configuration.

8 Follow the steps described in Installing and Monitoring a Single SQL Server Instance on page 11, starting with Step 5 to enter the monitoring credentials and enable the monitoring extensions for this instance.

> **NOTE:** For Amazon RDS SQL Server and Azure managed instance, the method of discovering instances by hosts or by IP range is unsupported in step 4.

> **NOTE:** To use stored credentials when enabling the Operating System monitoring extension, click the **Operating System > Select from stored credentials** link to open the Stored Credentials dialog box. Here you can review the log in credentials and authentication methods used for logging in to Foglight. Foglight stores encrypted credentials in lockboxes, which may be password-protected for added security. Database agents store all user log in credentials in a default lockbox called DB-Agent Lockbox. If credentials have already been entered in another lockbox, use the Lockbox list to select from that lockbox.

9 Click **Monitor**

The Monitoring progress bar appears.

At the end of this process, the Status column of the instance table displays either the status

Monitored for the instances that connected successfully to the database, or a status that indicates failure of the connectivity verification process and the reason for the failure. Click this text to view a dialog box that allows changing the credentials or to grant privileges, depending on the message that appears on the Status column:

- If the message is *Insufficient database privileges*, this issue can be resolved using the Insufficient Database Privileges dialog box that appears. For details, go to Step 10.

- For all other messages, the *Database Connection Failed* dialog box appears. For details, go to Step 11.

10 Click the text **Insufficient Database Privileges**.

The Insufficient Database Privileges dialog box appears.

This dialog box allows you to specify a SYSAdmin (System Administrator) user with sufficient privileges.

Enter a SYSAdmin user and password, and then click **Grant Privileges**. Alternatively, click the **View script** link, to the right of the **Grant privileges** button, to grant privileges manually with a script.

11 Click the message's text.

A dialog box that shows the message details appears.

a Click **Show details** to view the error's description.

b Click **Set credentials** to display the dialog box used for entering the credentials, and enter the required changes.

Adding instances manually

To add instances manually:

1 Click **Add instances**.

2 Click **Add manually**.

The Add New SQL Server Instance dialog box appears.

3 Use this dialog box to enter the following details:

- Server name — the SQL Server instance name. For default instance, specify the host name; for named instance, use the format: host name\instance name.
- Port — Required for a SQL Server instance whose TCP/IP connection port is other than the default port (1433).

IMPORTANT: If no port is specified, the wizard uses the port provided by the SQL Browser service (dynamic port), in which case this service must be started. After the instance monitoring configuration is complete, instances whose port was provided by the SQL Browser service have the number 0 displayed in their Port row.

4 Click **Finish**.

The newly added instance now appears on the table, with the status Set credentials.

5 Proceed with the wizard, starting from Step 5 on page 13.

Discovering instances by host name

Searching by host is the most efficient search method for discovering instances over a specified, small number of hosts.

To search by host:

1 Click **Add instances**.

The **Discover SQL Server Instances by Hosts** dialog box appears.

2 Type the host names in the field below the first check box. To specify multiple host names, separate the values by comma.

IMPORTANT: Ensure typing host names and not IP addresses. IP addresses are supported only when searching by IP range, as detailed in section Discovering instances by IP range on page 15.

Alternatively, if Foglight already monitors one or more hosts in the selected environment, select the requested host from the table of hosts currently known to Foglight.

IMPORTANT: The table lists all the hosts known to Foglight, regardless of the cartridge type, and including virtual hosts if the VMware cartridge is installed and configured.

3 Click **Discover**.

At the end of this process a table appears, displaying the newly discovered instances within the selected hosts.

4 Select the instances to be monitored.

5 Click **Finish**.

The newly added instance now appears on the table, with the status Set credentials.

6 Proceed with the wizard, starting from Step 5 on page 13.

Discovering instances by IP range

Searching by IP range is the recommended search method for discovering database instances over many hosts, by that means saving the need to specify the host names and ports.

To search by IP range:

1 Click **Add instances**.

2 Click **Discover by IP**.

The dialog box *Discover SQL Server Instance by IP Range* appears.

The IP range displayed by default is determined by the IP address and the subnet mask on the selected Foglight server.

3 In the **From** box, type the IP Range.

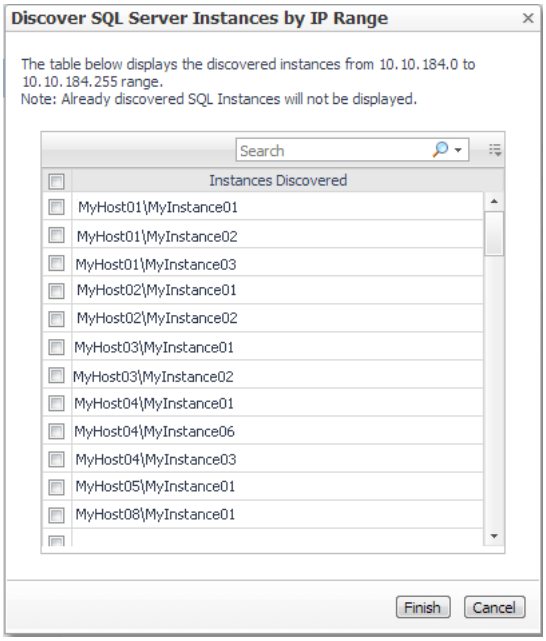
IMPORTANT: Ensure that the numbers entered are within the range of 0 to 255.

4 In the **To** box, type the requested IP Range.

IMPORTANT: Ensure that the numbers entered are greater than those entered in the From box, and that the address range does not exceed 1000 addresses.

5 Click **Discover**.

At the end of this process a table appears, displaying the newly discovered instances within the selected hosts.



6 Select the instances to be monitored.

7 Click **Finish**.

The newly added instance now appears on the table, with the status Set credentials.

8 Proceed with the wizard, starting from Step 5 on page 13.

Importing Instances from a File

Foglight for SQL Server supports importing an unlimited number of existing SQL Server instances that need to be monitored, using a predefined Regserv file that includes the properties of these instances.

IMPORTANT: The Regserv file can also be created by exporting the list of registered servers in SQL Server Management Studio. To do that follow the instructions at: https://msdn.microsoft.com/en-us/library/ms190246.aspx

To import the '<NAME>.regsrvr' file:

1. Select the **Import from file** option.
2. Import the `<NAME>.regsrvr` file
3. The instances listed in the file is presented in the SQL Instances list.
4. Select the instance you want to monitor and proceed with the validation step.

Configuring Multiple Instances for Monitoring using Silent Installation

To add multiple agents using the silent installation through a command line interface:

1. Navigate to **Administration** > **Cartridges** > **Components for downloads**.
2. Click the DB_SQL_Server_CLi_Installer name to download the installer.

   The Download Message dialog box appears.
3. Review the details in this dialog box, and click **Download**.

   The downloaded ZIP file contains the following files:

   - `README.txt` file
   - `mssql_cli_installer.groovy` — a groovy script file that runs the silent installation. This file should be copied to the <FMS_HOME>/bin directory.
   - `silent_installer_input_template.csv` — a template file that should serve as the basis for inserting contents into the input CSV file. This file can be copied to any folder of your choice, if the path indicated by the <`csv_instances_file_name`> parameter (see below) points to the selected path. For details about the contents of this file, see Contents of the Input CSV File on page 18.
   - `MSSQLPermissionsCheck.sql` — this file contains the SQL that the user should run on the monitored instance to check permissions needed for monitoring
   - `MSSQLPermissionsGrant.sql` — this file contains the SQL that the user should run on the monitored instance in order to grant the instance the required permissions for monitoring.
4. Extract the ZIP file to a directory of your choice.
5. Copy the `mssql_cli_installer.groovy` file to the <FMS_HOME>/bin directory.
6. Go to the command line and run the command: `<FMS_HOME>/bin/fglcmd -srv <fms_host_name> -port <fms_url_port> -usr <fms_user_name> -pwd <fms_user_password> -cmd script:run -f mssql_cli_installer.groovy fglam_name <fglam_name> instances_file_name <csv_instances_file_name>  lockbox_name <lockbox_name> lockbox_password <lockbox_password>`

   The descriptions of the flags and parameters in this command are as follows:

   - <FMS_HOME>: The Foglight Management Server installation directory.

- <fms_host_name>: Name of the host where the Foglight Management Server is installed.
- <fms_url_port>: The Foglight Management Server port.
- <fms_user_name>: The user name used for connecting to the Foglight Management Server.
- <fms_user_password>: The password of the specified user.
- <fglam_name>: The name of the selected FglAM where new agents are to be added.
- <csv_instances_file_name>: The full path and the name of the input CSV file.
- <lockbox_name>: Optional — specifies the name of an existing lockbox to be used.
- <lockbox_password>: Optional — the selected lockbox's password.

7 After the installation process completes, review the CSV files that are created in the same folder where the input file resides:

- A file with the input file's name and `_status` suffix (for example: if the input file is named `input`, this file is named `**input_new_-_**`).

  This file indicates the monitoring situation of all SQL Server instances listed in the input CSV file. If monitoring failed for one or more of the instances, the file specifies the error message.

  The `_status` file includes the name of the monitored Foglight for SQL Server agent, the result of the monitoring validation process — MONITORED, *FAILED*, or *AGENT EXISTS — and* the error message, in case the agent creation failed.

IMPORTANT: If all instances listed in the input CSV file passed the monitoring validation successfully, no additional file is created.

- A file with input file's name and `_new` suffix (for example: if the input file is named `input`, this file is named `input_status`).

  This file, which is only created if one or more instances failed the monitoring validation, contains all details about these instances.

  Using this CSV file, the user only needs to fix the errors and re-run the script, this time by specifying this file as the input file, that is: the original filename with suffix `_new`. In the example used here, the new file should be named `input_new_new.csv`.

Contents of the Input CSV File

The input CSV file contains the following fields, which are used as columns in the resulting file:

Table 1. Input CSV File

Name	Description
Host name	The SQL Server host name.
Instance name	The name of the SQL Server instance.
DB port (Optional)	The SQL Server instance port. If no value is inserted in this field, the port number should match the instance name.
DB credentials Type (Optional)	DB authentication type. The possible values are: • Windows_Default_Account • Windows_Custom_Account • SQL_Server_Authentication If this field is left empty, the Windows_Default_Account option is used.
DB user name (Optional)	The user name required for connecting to the database. If the *Windows_Default_Account* value was selected, this field can be left empty.
DB password (Optional)	The password required for connecting to the database. If the *Windows_Default_Account* value was selected, this field can be left empty.

Table 1. Input CSV File

Name	Description
DB power user credentials Type (Optional)	The type of database authentication of a user that can grant privileges to the regular database user. The possible values are: • Windows_Default_Account • Windows_Custom_Account • SQL_Server_Authentication. If this field is left empty, the grant privileges script will not run.
DB power user name (Optional)	The user name of a user that can grant privileges to the regular database user required only if the regular user does not have sufficient privileges.
DB power password (Optional)	The password of the database power user name mentioned above; required only if the regular user does not have sufficient privileges.
Should monitor OS	Specifies whether to create an Infrastructure Cartridge agent for the given host name; the possible values are either *True* or *False*.
OS credentials Type (Optional)	OS authentication type; can have one of the following values: • Windows_Default_Account • Windows_Custom_Account • Windows_Local_User: log in to Windows with the username and password running FglAM. • UX_Login_Password: log in to Unix through SSH with username and password. If this field is left empty, the Windows_Default_Account value is used by default.
OS user name (Optional)	The user name required for connecting to the OS if the Windows_Default_Account option was selected. The user name should be compatible with the selected authentication type. If the Windows_Default_Account value was selected, this field can be left empty.
OS password (optional)	The password of the OS user specified above; if the Windows_Default_Account option was selected, this field can be left empty.
Credential Name (Optional)	When using this property, the 'OS User Name', 'OS Password', 'Private Key File Path' and 'Passphrase' properties will be ignored.
Enable VMware Collection (Optional)	Specifies whether to monitor VMware metrics; the possible values are either *True* or *False*.
VMware Host (Optional)	The VMware host name.
VMware Port (Optional)	The VMware port. If no value is inserted in this field, the port number will use the default port number 443.
VMware User Name (Optional)	The user name required for connecting to the VMware.
VMware Password (Optional)	The password of the VMware user specified above.
Enable PI (Optional)	Indicates whether to enable PI; the possible values are *True* or *False.* Leaving this field empty indicates a *False* value. A PI repository must be already installed and assigned to at least one agent on the monitoring FglAM.

Table 1. Input CSV File

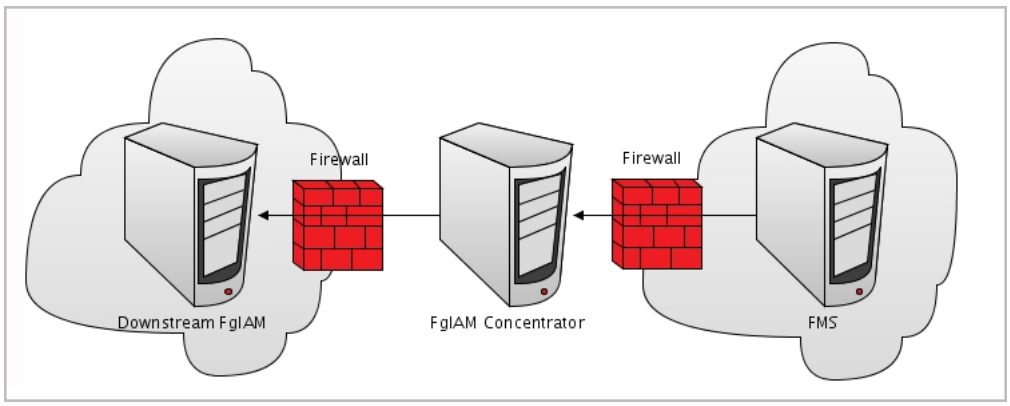
Name	Description
Is Mounted (Optional)	Indicates whether the given monitored instance is in a mounted stand-by mode (data-guard); the possible values are *True* or *False*. Leaving this field empty indicates a *False* value.
Group Names (Optional)	The database group names you want to add the monitored instance to; Multiple group name should separate with ";". For example: groupA;groupB;groupC When the group already exists, then the monitored instance will be attached to the existing group. When the group does not exist, then new group will be created under the default group "Database Groups".

Configuring the On Demand Data Port on the Agent Manager Concentrator

If a firewall is installed between the Management Server and the Agent Manager, a bidirectional On-Demand Data Port connection is not possible because On-Demand data requests require direct connection.

This issue can be resolved by creating an On-Demand Data Port on the concentrator — an Agent Manager instance that works similarly to an HTTP proxy. This On-Demand Data Port can be configured to accept On-Demand data requests from the Management Server and direct them to the Agent Manager instances (called downstream instances). For additional details about the concentrator, see Configuring an Agent Manager Instance as a Concentrator in the *Agent Manager Installation Guide*.

Figure 3. On-Demand Data Port connection with firewalls installed.



A concentrator agent, created specifically for Foglight for SQL Server, allows controlling the On-Demand Data Port so that the server is initialized (starts collecting data) when the agent starts, and stops when data retrieval by the agent ends. The agent, which the user creates manually, requires configuring the following parameters:

- A list of the downstream hosts that are connected through the concentrator
- The RMI port number

Each time the Foglight Management Server sends RMI request, the list of downstream hosts submitted by the concentrator agent is being searched, as follows:

- Inclusion of the target RMI server host name within this list indicates that a FglAM concentrator exists, in which case a connection is made to the proxy RMI server.
- If the list does not contain the target host name, the connection is made directly to the target On-Demand data port server.

To create the concentrator agent:

1 Go to **Dashboards > Administration > Agents** > **Agent Status**.

2 Click **Create Agent**.

3 Select the agent type **DB_SQL_Server_Concentrator**.

4 Enter a name of your choice.

5 Click **Create**.

6 After the creation process is completed successfully, click **OK**.

To edit the concentrator agent connection details:

1 Select the newly created concentrator agent.

2 Click **Edit Properties**.

3 Click **Modify properties for this agent only**.

4 Enter the number of the proxy On-Demand Data Port server port, or accept the default number.

5 Enter the concentrator Foglight Agent Manager host name (optional).

6 Click **Edit** to edit the list of downstream Agent Managers.

7 Click **Add Row**.

8 Enter the name of the requested Foglight Agent Manager host.

> **IMPORTANT:** When editing the properties of the Foglight Agent Manager concentrator (**Dashboards > Administration > Agents > Agent Status**), the name of the Foglight Agent Manager should be entered in the Downstream FglAMs section exactly as it appears in the topology, under **Home > Agents > All agents > <Agent name> > RMI data > FglAM host property**. The <Agent name> parameter refers to the Foglight for SQL Server agents that reside on the selected Foglight Agent Manager.

9 Repeat Step 7 to Step 8 as many times as required.

10 Click **Save Changes**.

11 Click **Save** at the lower right corner of the screen.

12 Click **Back to Agent Status**.

13 Select the newly created concentrator agent.

14 Click **Activate**.

15 Click **OK** to complete the process.

In an environment that includes a Foglight Agent Manager concentrator, when upgrading the environment where the concentrator does not contain any regular agents, the concentrator is not upgraded. Therefore, after the upgrade process takes place the Foglight Agent Manager concentrator has to be deployed manually.

2

Using Foglight for SQL Server

Foglight for SQL Server monitors the SQL Server database activity by connecting to and querying the SQL Server database. The agents provided monitor the SQL Server database system. The dashboards included with the cartridge provide a visual representation of the status of the major components of the SQL Server agents. They allow you to determine any potential bottleneck in database performance.

Viewing the Databases Dashboard

> **NOTE:** The following section describes the various components of the Databases dashboard. To learn how to customize this dashboard's display and focus on a requested instance, see Selecting an Instance to Monitor on page 24.

The Instance home page is opened by accessing the Databases dashboard.

To access the Databases dashboard:

1. Ensure that the navigation panel on the left is open.

   To open the navigation panel, click the right-facing arrow on the left.

2. On the navigation panel, click **Homes > Databases**.

The Databases dashboard provides an at-a-glance view of the monitored environment, with all the currently monitored database types.

The Databases dashboard includes these sections:

- Database cartridge type tiles — each tile represents a database type (SQL Server, SQL Server BI, Oracle, Sybase, DB2, MongoDB, Cassandra, or All Instances). Tiles display the number of instances for each database type, along with a breakdown according to the instance health (Normal, Warning, Critical, Fatal, or Unknown).
- Status section — includes the following components:
  - **Status summary** — a color-coded bar, which provides a visual representation of the summarized health condition of all instances listed in the Database Group table.

    The status summary bar provides a graphic representation of the monitored environment's current state, broken down to the number of instances and their current health state: Fatal, Critical, Warning, Normal or Unknown.
  - **Database-specific health summary** — when the database group All is selected, this section displays all the currently monitored instances for each database type, divided by their health state (for example: four SQL Server instances, three of which have the health state Warning and one is indicated with the health state Fatal). When a user-defined database group is selected, this section displays data only about the agents included within the selected group.

  > **NOTE:** The Status Indicators section can be used for filtering the Databases Group table to display only instances that meet a criterion set in this section. For details, see Filtering the Display by Severity on page 25.

- The **Monitor** button — Use this button to add instances to monitor. For details, see Monitoring Data Replication on page 35.

- The **Configure Alarms** button — Takes you directly to the Administration > Alarms page. On the Alarms page you can view the alarms as well as configure the Alarm Template and alarm notification settings.
- The **Settings** button — Use this button to do one of the following:
  - Access the User Management settings, allowing you to restrict which instances specific users are allowed to view. This makes it easier to for users to find information about only the instances they are interested in. For details, see Assigning Instances to Users or Groups on page 23.
  - After selecting one or more instances of the same database type, use this button to set options for collecting, storing, and displaying data for the selected instances, and configuring the connection to SQL Performance Investigator (if installed). For details, see Viewing User-Defined Collections on page 132.
- **Currently selected database group table** — a list of all monitored databases within the database group that is selected in the Databases section. For details, see Currently Selected Database Group on page 24.
- The **Select dashboards ( )** button — provides direct link to several drilldowns and panels, by that means saving the need to navigate to these locations through the Overview drilldown.

Assigning Instances to Users or Groups

The User Level Access screen allows you to assign specific instances to users or groups. When users view their Databases dashboard, they only see the instances assigned to them.

NOTE: The Status Indicators section filters the Databases Group table to display only instances that meet a criterion set in this section. For details, see Filtering the Display by Severity on page 25.

The User Level Access screen displays all users and groups as they appear in the Foglight User & Security dashboard. By default each user is assigned to view All instances monitored in the Foglight Management Server.

To assign instances to a specific user or group:

1. On the main Database dashboard, click **Settings** and select **User Level Access** from the menu.

   The Users and Groups panes appear.
2. To assign instances to a specific user or group, locate the user name in the table. Click the relevant value under the required domain in the Assigned instances column. You can also click **Configure** in the right column.
3. Select one of the 3 options:
   - **Manage** — allows you to manage which instances or groups to view
   - **All Instances** — set the user to view All monitored instances
   - **None** — block the user from viewing any monitored instances
4. Select **Manage the instances** to display the Assign Instances view.
   1. Select the instances or the pre-defined groups of instances.
   2. Click the arrow button to move the instances from the Assigned Instances box to the Available instances box.
5. Click **Set** save your choices and exit.

NOTE: A Reset button is available in the Users table toolbar. This button allows you to reset user level access to the default behavior. All users will be assigned to view all instances.

Selecting an Instance to Monitor

Because the Databases dashboard displays by default all the currently monitored databases, customize the dashboard's view to display only the relevant instances, and then preview such instances to decide which to monitor.

To select an instance to monitor:

1 Display only instances relevant for your needs, using one of the following methods:

- **Filter by database type** — if multiple database types are being monitored, click the database type tile that represents the requested type (in this case, SQL Server).
- **Filter by severity** — use the status indicators to display only instances of a specific database type that share a specific health state severity. For further details, see Filtering the Display by Severity on page 25.
- **Create user-defined groups** — use the Databases area to create groups that contain only the databases that need to be monitored for a specific need. For further details, see Creating Custom Database Groups on page 25.
- **Assign Instances to specific users** — When accessing a Database dashboard, non-administrative users view only instances which have been assigned to them. For details, see Assigning Instances to Users or Groups on page 23.

2 View the selected instances' severity level, using the status indicators.

3 Click the requested instance to view a cue card with a preview of the instance's most significant performance indicators. For details, see Previewing the Requested Instance on page 27.

4 Click **Home Page** on the cue card to open a full-screen view of the requested instance.

IMPORTANT: Launching a full-screen view of the requested database by clicking the **Home Page** link can be carried out only for instances monitored in Foglight for SQL Server mode.

Currently Selected Database Group

This Databases table displays the group that is selected from the available database groups under the Databases section of the navigation pane.

The Databases table contains the following columns:

NOTE: Because the currently selected group can contain instances of types of database monitored by Foglight, the table describes instances in general.

Table 2. Databases Table

Name	Description
Database	
Sev	The instance's highest severity alarm, which determines the instance's health state.
Name	The name of the instance.
Version	The version number of the instance.
Up Since	The date and time when the instance was last restarted.
Workload	Displays the workload chart for the instance.

Table 2. Databases Table

Name	Description
DB Alarms	The number of warnings, critical, and fatal alarms for the instance. Shows the most recent alarms invoked for the specific instance displayed in the row. The alarms are displayed by their severity levels, with the aggregated number for each severity. Clicking the alarm's severity icon displays the Alarm pop-up, which provides in-depth information about the alarm, its causes and implications.
System Utilization	
Host	The name of the computer that hosts the instance.
CPU Load (%)	The overall operating system CPU usage (including CPU usage by the database).
Memory (%)	The percentage, within the total memory, of memory consumed by all operating system processes (including the database). This value includes both RAM resident memory and swapped memory.
Disk (% Busy)	The percentage of time the busiest device spent serving system-wide input and output requests. This metric serves as a measure for the system I/O load.
Monitoring Status	
Agent	The operational status of the monitoring agent. When the agent instance is running, hovering over the Stat icon displays the status message Collecting Data. When the agent instance is running but not collecting data, hovering over the State icon displays one of the following status messages: • Starting • Stopped • Stopping • Unknown
OS	The state of OS-related data retrieval by the database cartridge, the Infrastructure cartridge, or both.

Filtering the Display by Severity

The Status Indicators section can be used for filtering the Databases Group table to display only instances that meet a criterion set in this section.

To filter the display by severity:

1 Click a specific type of severity within a specific database type (for example, SQL Server instances whose severity level is Fatal).

The selected filter is shown in the table title.

2 To select another filter, click the required status indicator (for example, SQL Server > Normal severity). To display the entire list of databases, click **Database type tile**.

Creating Custom Database Groups

While all existing and newly discovered instances are added to the All list, the Databases area of the navigation pane allows creating, editing, and removing sub-groups of database instances, by that means serving as a filter that allows displaying only a specific, user-defined group of instances.

The Databases area includes the following components, which can be selected from a drop-down list:

- Database Groups — includes by default the All list, and the Database Groups list, which serves as the parent group for creating user-defined sub-groups of database instances. Such sub-groups are recognized as Foglight services.

  The Database Groups section also displays the highest severity level of the database groups, under the column **Sev**.

- Services — displays the configured services, which contain at least one object (database topology) that Foglight for SQL Server uses.

  All the Foglight for SQL Server sub-groups, which were created using the Database Groups section, are displayed also under Services. In addition, Services displays groupings of monitored resources that were created using the Service Builder dashboard, and contain Foglight for SQL Server database topology.

  The Services section also contains the column SLC (Service Level Compliance), which indicates the current availability of the selected service over a given period.

  For details about the Service Builder dashboard, see the Foglight Online Help.

To create a database group:

1 Select the parent database instance group, Database Groups, under Databases in the navigation pane.

2 Click the button.

The Add Sub Group dialog box appears.

3 Type a name for the group in the Name field.

4 Type a description for the group (optional).

5 Select an instance in the Available column, and click [>] to move the instance to the Selected column.

Alternatively, click [>>] to move all the databases to the Selected column.

6 Click **OK**.

The sub-group name appears in the Database Groups list.

The database instances are listed in the Databases dashboard.

After adding one or more user-defined sub-groups, it is possible to add sub-groups to these sub-groups.

To remove a database sub-group:

1 Select the sub group to remove.

2 Click the button.

The verification dialog box appears.

3 Click **Remove**.

The sub-group is removed from the Database Groups list.

To edit a database sub-group:

1 Select the requested sub group.

2 Click the button.

The Edit Group dialog box appears.

3 To add an instance to the sub group, select the requested instance in the Available column and click [>].

The selected instance moves to the Selected column.

4 To remove an instance from the sub-group, select the requested instance in the Selected column, and click [<].

The selected instance moves to the Available column.

5 Click **OK**.

Previewing the Requested Instance

Clicking the eye icon next to the instance name opens the quick view panel. The quick view provides a preview of the instance's most significant indicators, using the following sections:

- General — identical to the home page's Instance pane.

  This section provides the following details:

  - DB Type — identifies whether the database type of the monitored instance is SQL Server, Sybase, DB2, or Oracle.
  - Version — identifies the database version number, along with the most recently installed service packs.
  - Up Since — identifies the date and time when the SQL Server instance was last started.
  - Host — identifies the host's name.
  - OS Cluster — indicates whether the instance is part of an OS cluster.
- Alarms — displays the number of warning, critical, and fatal alarms for the instance.
- SQL Performance — opens the History drilldown, which displays SQL performance-related data, either at a high-level or in-depth. This option is available only for instances that have SQL PI configured.
- HADR — indicates whether one or more of the following services are configured:
  - Mirroring — points out whether one or more databases within the monitored instance take part in a mirroring operation, either as a principal database whose exact copy is mirrored on a different instance, or as a mirror database. For details, see Tracking the Status of the Mirroring Operation on page 120.
  - Log shipping — used for setting up back-up databases to take the place of a current "live" database if that database goes down. For details, see Tracking the Status of the Mirroring Operation on page 120.
  - Replication — for details, see Configuring the On Demand Data Port on the Agent Manager Concentrator on page 20.
  - Always On — for details, see Reviewing the Always On Availability Groups on page 124.
  - Cluster Aware — indicates whether the instance is part of a SQL Server cluster.
- Host — displays the following indicators about the operating system's performance:
  - CPU load
  - Memory utilization
  - Disk utilization
- Storage — displays the storage's total percentage of used space, and the following indicators:
  - Total number of file groups
  - Total number of files
  - Total volume of occupied disk space, in gigabytes
- Workload — the workload (average active sessions) graph for the SQL Server instance, plotted over the specified time range (by default: last 60 minutes).

  The Workload section also contains the following buttons:

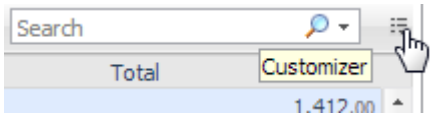
  - Overview — allows accessing the Foglight for SQL Server Overview page.

- Replication — this button, which is enabled only when the monitored instance is detected as a distributing instance, allows accessing the Replication home page. For more information, see Configuring the On Demand Data Port on the Agent Manager Concentrator on page 20.
- SQL Performance — opens the SQL Performance drilldown, which displays performance-related data, either at a high-level data for all instances or in-depth data, available only for databases which are configured with SQL Performance Investigator. For details, see Monitoring SQL Performance on page 46.

Components Shared by all Foglight for SQL Server Screens

The tables' Customizer button and the In-context action button are common to all Foglight for SQL Server screens, drilldowns, and panels.

Figure 4. Customizer button



- The **Customizer** button — depending on how the chart was designed, the customizer icon might appear when you hover the cursor over a chart or might always be visible. For example, on the Alarms dashboard, the customizer for the Alarm(s) table is always visible. If the customizer is enabled, you can
  1. Select which columns you want to appear in a table.
  2. Change the chart type dynamically. See **Changing the Chart Type**.
  3. Export charts and tables to PDF, CSV, Excel, XML, or Image format. See **Exporting Data from Charts and Tables**.
- In-context actions button — found on the upper right corner of all screens. On selecting it a menu displays allowing actions such as refresh of the viewed screen, quick access to the Agent Admin dashboard and shortcuts to documentation info and support."

Foglight for SQL Server Toolbar and Views

The Foglight for SQL Server toolbar provides access to detailed views you can access using the toolbar.

Breakdown and Baseline Chart Formats

Several charts can display information in either of the following formats:

- Baseline format — where a selected individual metric is displayed as a single line.
- Breakdown format — a representation of actual activity of a metric or a set of metrics, compared with the typical behavior of these metrics for the selected time range. A breakdown display can also present a single metric divided by various components (for example, space utilization divided by the various components that occupy this space).

Home Page Toolbar

The Foglight for SQL Server Overview page provides the following toolbar buttons:

Table 3. Options on the Overview page.

Name	Description
Overview	Opens the Overview Dashboard.
Advisories	Opens the Advisories Dashboard.
SQL PI	Opens the SQL Performance History drilldown, which displays SQL performance-related data at the following levels: • High-level data — delivered through the History view or the Lock Analysis view. For details, see Viewing Metrics on page 50 and Blocking History on page 50 • In-depth data — available only for instances that have SQL PI configured. The SQL Performance drilldown contains the following panes: ▪ Investigate — see Monitoring SQL Performance with SQL PI Configured on page 53 ▪ History — see Viewing Historical Metrics on page 54 ▪ Change Tracking — see Viewing Change Tracking on page 59 ▪ Lock Analysis — see Blocking History on page 50
Memory	Opens the Memory drilldown described in Reviewing Memory Usage on page 62. The Memory drilldown contains the following panels: • Summary — see Viewing the Memory Summary on page 62. • Buffer Cache — see Monitoring Plan Cache-related Data on page 66. • Plan Cache — see Monitoring Plan Cache-related Data on page 66.
Activity	Opens the Activity drilldown described in Reviewing the Instance Activity on page 69. This drilldown contains the following panels: • Real-time Summary — see Viewing the Real-time Summary Page on page 73 • SQL Instance Summary — see Reviewing the SQL Server Instance Activity on page 69. • SQL I/O Activity — see Viewing SQL I/O Activity Data on page 73. • Sessions — see Reviewing Session Details on page 86. • Locks — see Monitoring Locks and Latches on page 92. • Blocking (Current) — see Tracking Current Lock Conflicts on page 93. • Deadlocks — see Tracking Deadlocks and their Affected Objects on page 94. • I/O by File — see Viewing I/O Statistics by Database Files on page 97.
Databases	Opens the Databases drilldown which contains the following panels: • Summary — described in Reviewing Database Usage on page 98. • TempDB — described in Monitoring TempDB Status on page 112.

Table 3. Options on the Overview page.

Name	Description
Services	Opens the Services drilldown described in Reviewing the Services on page 112. This drilldown contains the following panels: • Services Status — see Reviewing the Support Service Status on page 113. • SQL Agent Jobs — see Reviewing SQL Agent Jobs on page 114. • SQL Agent Alerts — see Viewing SQL Agent Alerts on page 115. • DTC — see Monitoring SQL Server Transactions using the DTC Panel on page 116. • Full Text Search — see Monitoring Full-text Indexes using the Full Text Search Panel on page 117. • Reporting Services — see Monitoring Reports using the Reporting Services Panel on page 136.
HADR	Opens the HADR drilldown, described in Using the HADR Drilldown on page 117. This drilldown contains the following panels: • Log Shipping — see Monitoring the Log Shipping on page 118. • Cluster Services — see Monitoring Cluster Services on page 119. • Mirroring — see Tracking the Status of the Mirroring Operation on page 120. • Always On — see Reviewing the Always On Availability Groups on page 124.
Logs	Opens the Logs drilldown described in Using the Logs Drilldown on page 129. This drilldown contains the following panels: • SQL Server Error Logs — see Reviewing the SQL Server Error Logs on page 129. • SQL Agent Error Logs — see Reviewing the SQL Agent Error Logs on page 129.
Configuration	Opens the Configuration drilldown described in Reviewing Configuration Settings on page 130. This drilldown has only one panel, SQL Server Configuration. Reviewing SQL Server Configuration on page 130
User-defined	Opens the User-defined drilldown described in Viewing User-defined Performance Counters and Collections on page 131. This drilldown contains the following panels: • Performance Counters — see Viewing User-Defined Performance Counters on page 131. • Collections — see Viewing User-Defined Collections on page 132.

Overview Dashboard

The upper section of the dashboard includes general information on the monitored instance:

- SQL server version
- Host. For clustered host, the active host is displayed.
- OS Version
- Instance Collation

Each tile is constructed of a title that state the name of the monitored issue and an aggregation of the alarms relevant to that issue.

The tiles are organized by priority:

- Availability — show information and alarms about the availability of the instance. Also includes the monitoring state for that instance.
- HA/DR — High Availability (HA) and Disaster Recovery (DR) state of the instance. Displays the state of the resources used for availability. For example, Always On, Cluster, Replication, Mirroring, Log Shipping.

  NOTE: Only the three resources would directly be displayed, according to the importance. The state of backup's alarms is also displayed.

- Storage — the space utilization level for all Data and Log file groups.
- Infrastructure — general information collected from the server:
  - CPU — the SQL Service CPU level of utilization in compare to the general utilization of the SQL Service.
  - Memory — the amount of memory the SQL Service uses compared to other processes and the total memory on the server.
  - Disk I/O — Displays the utilization of the busiest disk on the monitored server.
  - Virtual Overhead — Shows (only if the virtual machine is monitored) the overhead (In CPU) of the server on the hosting machine.
- Operational — displays:
  - Alarms on failed jobs.
  - Alarms for jobs that have exceeded their average duration.
  - Alarms on the error log.
  - Alarms related to other issues. For example, agent log, CLR and more, are only displayed at the top of the tile.
- Performance — summarizes general information collected from the SQL Performance dashboard. The pie chart displays the average workload divided by resources over time. The graphs display the total workload trend over time.

  The Instance Performance tile also displays the following issues:
  - Throughput issues — for example, average batches, logons and transactions per second.
  - Current sessions state — for example, Active, Inactive and Blocked.

The right panel is used to display either alarms or Top 10 SQLs:

- Alarms
  - Display all active alarms for the underlined instance.
  - Enable sorting by creation time or severity.
- Top 10 SQLs displays the 10 SQLs with the longest duration.

Advisories Dashboard

The *Advisories* dashboard is provided to continually analyze application performance to identify performance inefficiencies, to guide you through problem resolution strategies, and to deliver a step-by-step action plan for maximizing database performance.

The upper section of the *Advisories* dashboard includes general information on the monitored instance:

- SQL server version
- Host. For clustered host, the active host is displayed.
- OS Version
- Instance Collation

The lower section of the *Advisories* dashboard contains the following two elements:

- *Action Plan*: Summarizes available advisories that present opportunities to increase the overall performance of your database.
- *Advisories*: Lists advisories, action types, and the relevant descriptions. Click each advisory to view detailed analysis results, a complete description of the recommended action to be taken, and the background information of this tuning area.

  TIP: Advisories are listed in the order of their priority that considers the severity of the detected deviation and the type of advisory.

Refer to the table below for the Lists of advisories and short descriptions of their purpose:

General terms used in the Advisory List:

- Active Time: Active time measures the amount of seconds all sessions inside connected to the instance used during their activity. This activity includes both using CPU and waiting for various resources like Locks, IO, etc. When the number gets bigger, it can indicate more sessions where connected, more waits were encountered, or both. Active Time rate shows how many activities were done each second on average and it is roughly correlate to the number of threads that were active.
- Baseline: Normal range for a metric over a specific point in time.

Table 4. Foglight for SQL Server Advisory List

Advisory Name	Description
Overall Resource Consumption Exceeds the Baseline	Indicates that the database system's performance during the specified time frame, exceeded the normal rate relative to a calculated baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
CPU Usage Deviation	Indicates that the total instance CPU usage exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Calculated every 1 hour.
I/O Wait Deviation	Indicates that the total time spent waiting for I/O requests to finish, exceeded the baseline. I/O wait includes both waits for physical I/O operations and logical I/O (waits to read from the buffer pool). Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
Memory Wait Deviation	Indicates that the total time spent waiting on various wait types related to Memory, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Calculated every 1 hour.
Network Wait Deviation	Indicates that the total time spent waiting on various wait types related to Network, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.

Table 4. Foglight for SQL Server Advisory List

Advisory Name	Description
Lock Wait Deviation	Indicates that the total time spent waiting for locks to be released, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Calculated every 1 hour.
Latch Wait Deviation	Indicates that the total time spent waiting on various wait types related to Latch, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
Log Wait Deviation	Indicates that the total time spent waiting on various wait types related to Log, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
CLR Wait Deviation	Indicates that the total time spent waiting on various wait types related to CLR, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Calculated every 1 hour.
Remote Provider Wait Deviation	Indicates that the time spent waiting on various wait types related to Remote Provider, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
Other Wait Deviation	Indicates that the total time spent waiting on waits categorized as “Other”, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
XTP Wait Deviation	Indicates that the total time spent waiting on waits related to in-memory OLTP, exceeded the baseline. Deviations outside the baseline range suggest abnormal or unexpected activity and should be investigated to determine whether they suggest a performance problem. Analyzed every 1 hour.
Table Experiencing Lock Contention	The advisory highlights tables and indexes that are in the source of lock issues affecting the instance. It helps to identify the objects in question, analyzes the effect of locks on the entire system, and provides details to resolve them. Analyzed every 1 hour.
Deadlock Observed	The advisory is raised when the number of deadlocks per minute is bigger than 1. It includes details about the database, application, and objects that were involved in the deadlocks occurred. Analyzed every 1 hour.

Table 4. Foglight for SQL Server Advisory List

Advisory Name	Description
Excessive Statements Re-compiles	The advisory is raised when the rate of recompiles compare to the amount of batches during the analyzed period is bigger than 10% and instance is CPU bound. Analyzed every 1 hour.
Buffer Latch Contention	The advisory is raised when the amount of waits for buffer latch is more than 10% of instance activity. It provides recommendations on how and what to tune to reduce it. Analyzed every 1 hour.
Excessive Memory Pressure	The advisory is raised when Foglight detects various symptoms that indicate memory pressure exists. It provides details on what those symptoms are and recommendations on how to fix it. Analyzed every 1 hour.
Excessive Log Wait	The advisory is raised when the amount of waits related to transaction log is more than 20% of instance activity. It provides recommendations on how and what to tune to reduce it. Analyzed every 1 hour.
Excessive I/O Wait	The advisory is raised when the amount of waits related to physical I/O is more than 5% of instance activity. It provides information about top statements consuming this I/O and details about missing indexes that may contribute to the situation. Analyzed every 1 hour.

Monitoring Data Replication

If the monitored SQL Server environment contains one or more instances that are configured as distributors (distributing instances) in a replication operation, Foglight for SQL Server now provides a replication monitoring module.

IMPORTANT: Replication is monitored for instances running on SQL Server 2005 and higher.

The Replication Available flag indicates if an instance is configured as a distributor in these ways:

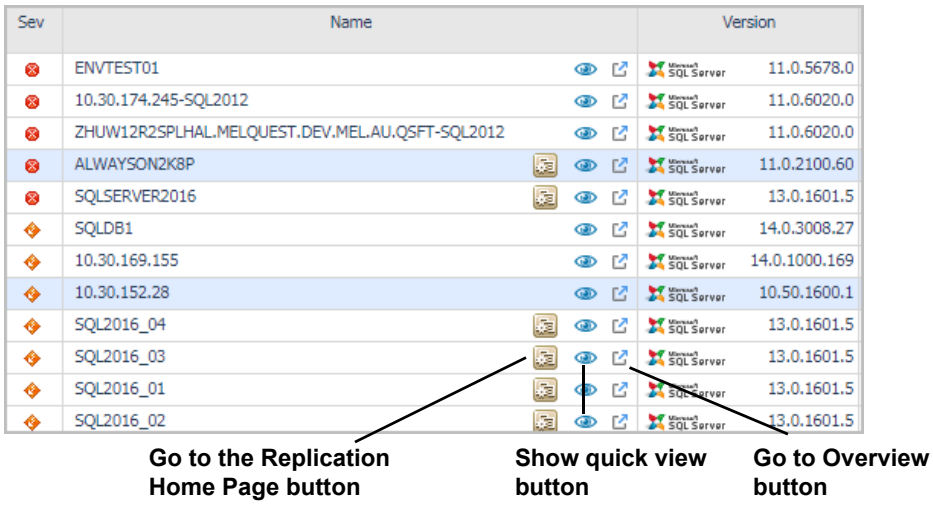
- After the instance has been discovered (if the configuration as a distributor took place before the instance was discovered)

Upon the configuration as a distributor for monitored instances. Instances detected as distributors appear with the button **Go to the Replication Home Page** ( ) next to the **Show quick view** button on the Name column of the Databases table.

The replication home page can be accessed using one of the following methods:

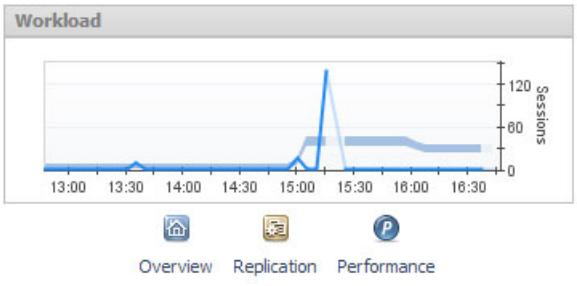
- Click the button **Go to the Replication Home Page**.

Figure 5. The Replication button in the Databases table.



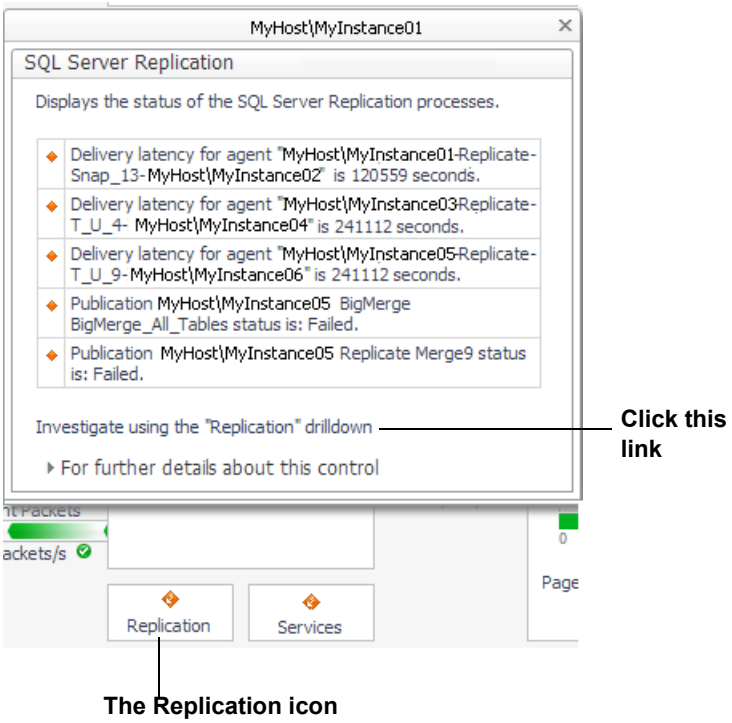
- Click the **Replication** button on the QuickView panel.

Figure 6. The Replication button in the QuickView panel.



- Click the link **Investigate via Replication dashboard** in the SQL Server Replication pop-up, which appears when you click the Replication icon under the Background Processes section of the instance's home page.

Figure 7. The Investigate using the Replication Drilldown link.



Viewing the Replication Home Page

The table identifies the main elements of the Foglight for SQL Server Replication Home page window, and provides a link to display more information, corresponding to the logical dataflow within the Replication Home page.

Table 5. The Main Elements of the Foglight for SQL Server Replication Home Page window.

Part Name	Description
Replication-related alarms	Monitoring Alarms on page 45
Publications drilldown	Monitoring the Replication Publications on page 39
Agents drilldown	Monitoring the Replication Agents on page 41
Subscriptions drilldown	Monitoring the Replication Subscriptions on page 43
Publishing pane	Reviewing the Publishing Process on page 37
Distributor Agents pane	Reviewing the Distributor Agents on page 37
Subscription pane	Reviewing the Subscription Process on page 38
Data flow indicators	Reviewing Data Flow Indicators on page 38

The following provides a description of panes of icons and gauges in the main activity areas:

- Instance identification — used for identifying the instance, its type, and operation time. Identical to the instance identification section of the home page. For details, see Identifying the Instance on page 74.
- Replication-related alarms — displays only the alarms invoked upon deviations from predefined thresholds of replication-related rules. For details, see Monitoring Alarms on page 45.
- Drilldowns — used for carrying out in-depth investigation of each of the main components of the replication process: publications, agents and subscriptions. Access drilldowns by clicking their button on the toolbar or the relevant link on their corresponding panes.

- Components representing instance data flow — the main activity area in the Foglight for SQL Server Replication Home Page includes several flows that represent the data flow in the replication process. The data flow is described in this topic in a top-down design from the Publishing process to the Subscription process.
- Panes — displays summarized numerical data about each of the components that take part in the replication-related processes of publishing, distributing, and subscribing. Clicking each component allows you to view a detailed list of the components (for example, Publisher servers). You can also access the relevant drilldown (for example, Publication Replications) for viewing more detailed information about the relevant component in the replication process.

Reviewing the Publishing Process

The *Publishing* pane provides data regarding the following components of the publishing process:

- Publishing servers — displays the total number of publishing servers (Publishers) that are currently distributing publications using the selected instance. This number includes publishing servers that reside on the same computer as the distributing instance, or that reside on remote computers.

  Clicking this row displays a pop-up, which contains the Publishing servers and Publication databases tables. Each of these tables shows the displayed entity's name, status and highest severity level.

  NOTE: If more than one publishing servers exist, clicking each publishing server on the table displays only the publication databases residing under the selected server.

- Publication databases — displays the total number of publication databases that reside under all currently distributing publishers.

  Clicking this row displays a pop-up containing a table of all publication databases. This table is identical to the one shown when clicking the *Publishing server* row, except for the filtering by server, which is carried out by clicking the **Server** column.

  IMPORTANT: Clicking each publication database's row opens the *Replication Publications* drilldown, which only displays publications stored on the selected publication database. To view the entire list of publications from all publication databases, click the **Publications** toolbar button.

- Publications — displays the total number of publications that are currently distributed using the selected instance, along with an icon that indicates the highest severity level.

  Clicking this row opens the *Replication Publications* drilldown, which allows reviewing publications by their type. For details, see Monitoring the Replication Publications on page 39.

Reviewing the Distributor Agents

The Distributor Agents pane displays the following replication agent types, indicating for each type the number of agents and the highest severity level:

- Merge — in merge replications, the Merge job first applies the initial snapshot to the subscribing server (Subscriber). Then it connects to the publishing and subscribing servers to update both sides with each incremental data change that takes place.
- Distribution — in both transactional and snapshot replications, the Distribution job applies the initial snapshot to the subscribing server. In transactional replications, this job also transfers (downloads) newly added transactions stored in the distribution database to the subscribing server.
- Queue Reader — in transactional replication with the queued updating option, this agent moves (uploads) changes made at the subscribing server back to the publishing server, by reading messages stored in a Microsoft SQL Server queue or a Microsoft Message Queue. Then it applies those messages to the publishing server.

  IMPORTANT: Unlike the Distribution agent and Merge Agent, the Queue Reader agent is an optional component. Only one Queue Reader Agent job exists to service all publishing servers and publications for a given distribution database.

- Snapshot — in all replication types, the Snapshot agent, which runs on the distributing instance, prepares and maintains the snapshot and synchronization operation. This agent prepares the snapshot files, which contain the schema and data of the entities that are to be published (both tables and database objects). The agent stores the files in the snapshot folder, and tracks synchronization status by recording the synchronization jobs in the distribution database.
- Log Reader — in transactional replications, the Log Reader agent monitors the transaction log of each published database. It copies the transactions marked for replication from the log to the distribution database.
- Misc. Jobs — various replication maintenance jobs, which are automatically created by Microsoft.

  NOTE: Some of the miscellaneous jobs are not started by default. In which case a Warning icon appears near the job's name, and its status is Not Started. It may be advisable to start such a job manually at some point; for example, to clean up the distribution tables.

Selecting a specific agent type's row (for example, Merge) and clicking it opens the Replication Agents drilldown, displaying only the selected agent type. For details, see Monitoring the Replication Agents on page 41.

Reviewing the Subscription Process

The *Subscription* pane provides data regarding the following components of the subscription process:

- Subscribing servers — displays the total number of subscribing servers (Subscribers) that are currently making data available to other locations through replication, by hosting one or more Subscription databases. This number includes both subscribing servers that reside on the same computer as the distributing instance and subscribing servers that reside on remote computers.

  Clicking this row displays a pop-up, which contains the *Subscribing servers* and *Subscription databases* tables. Each of these tables shows the displayed entity's name, status and highest severity level.

  NOTE: If more than one subscribing servers exist, clicking each subscribing server on the table displays only the subscription databases residing under the selected server.

- Subscription databases — displays the total number of subscription databases that reside under all currently subscribing servers.

  Clicking this row displays a pop-up that contains a table of all subscription databases. This table is identical to the one shown when clicking the *Subscribing server* row, except for the filtering by server, which is carried out by clicking the **Server** column.

  IMPORTANT: Clicking each subscription database's row opens the *Replication Subscriptions* drilldown, which only displays subscriptions stored on the selected subscription database. To view the entire list of subscriptions from all subscription databases, click the **Subscriptions** toolbar button.

- Subscriptions — displays the total number of subscriptions that are currently receiving selected publications through the selected instance, along with an icon that indicates the highest severity level.

  Clicking this row opens the *Replication Subscriptions* drilldown, which allows reviewing subscriptions by their type. For details, see Monitoring the Replication Subscriptions on page 43.

Reviewing Data Flow Indicators

The data flow indicators provide a representation of the average delivery rate of each of the replication agent types, except for Misc. Jobs. A numerical value indicates the average number of commands transferred using the specified agent during the specified time range.

Clicking each flow displays a pop-up that contains a description of the flow, and a chart that shows the relevant data plotted over the specified time range.

Monitoring the Replication Publications

The Replication Publications drilldown allows viewing more detailed information about all publications distributed using the selected instance.

To investigate the requested publication:

1 Select which publication type to display, by selecting the tile that represents the requested type: Snapshot, Transactional, Transactional with update, Merge or All.

> **IMPORTANT:** In each of these tiles, it is possible to filter the display to show only publications with a specific severity status, by clicking the icon that represents that status. For example, Not Started.

2 Review the requested publication data on the Publications table:

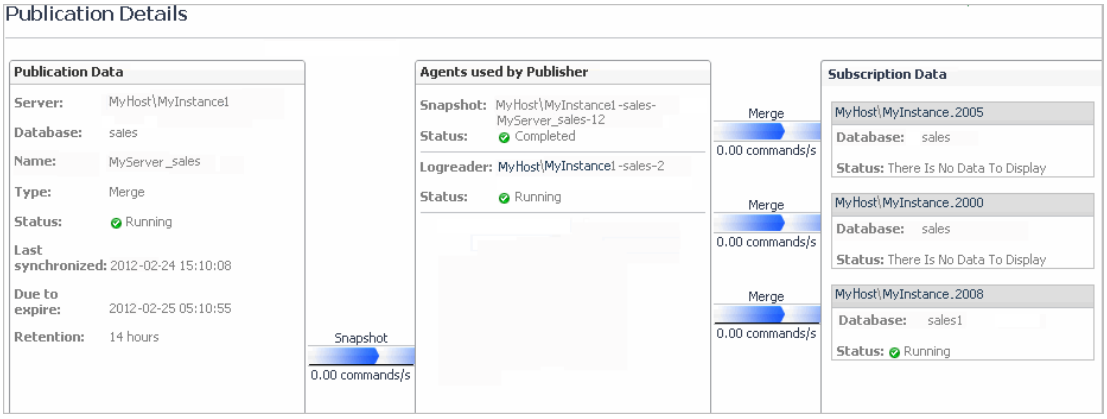
- Severity — the highest severity level
- Status
- Publishing server name
- Publication database name
- Publication type
- Number of subscriptions
- Agents taking part in the transaction — Log Reader, Queue Reader, and Snapshot agents

3 Click the publication's row to access the **Publication Details** screen.

The Publication Details Screen

The Publication Details screen allows tracking the process of publishing the selected publication, by displaying all components involved in the process and providing links for further investigation.

Figure 8. The Publication Details screen.



Similar to the Replication home page, this screen contains the following panes, each representing a stage at the replication process:

- Publication Data — provides the publication-related data that appears on the Replication home page, that is, publishing server, publication database, and publication name, and the following data:
  - Type — the publication type (transactional, snapshot, and so on).
  - Status — can have one of the following values: *Running*, *Failed*, *Retrying*, *Idle*, and *Not started*.
  - Last synchronized — the time and date when the publication data items residing on the subscribing and publishing server were most recently synchronized. This parameter is highly significant for publications configured to run continuously.

- Due to expire — displays the time and date when the subscription expires, if changes recorded in the distribution database are not yet synchronized with the subscribing server.

NOTE: The value displayed on the *Due to expire* field is set by adding the number of hours defined in the Retention value to the value displayed in the *Last synchronized* field. For example, if the most recent synchronization took place on 17:00, and the retention period is 14 hours, the value displayed here is 7:00.

- Retention — indicates the predefined distribution retention period. If by the end of this period there are changes in the distribution database that have not been delivered to the subscribing server and synchronized, the subscription expires and drops.

- Agents used by Publisher — displays the replication agent used for replicating the publication data to the subscribing server.

  The agent types displayed on this pane for each publication type are as follows:

  - In snapshot, Transactional and Merge publications — Snapshot and Log Reader.
  - In *Transactional with update* publications — Snapshot, Log Reader and Queue Reader.

  IMPORTANT: To view a detailed representation of the agent's actions using the Replication Agents drilldown, click the requested agent's row. The Replication Agents drilldown appears, with the clicked agent's row selected in the All Agents table. For details, see Monitoring the Replication Agents on page 41.

- Subscription Data — provides the following data regarding each of the publication's subscriptions:

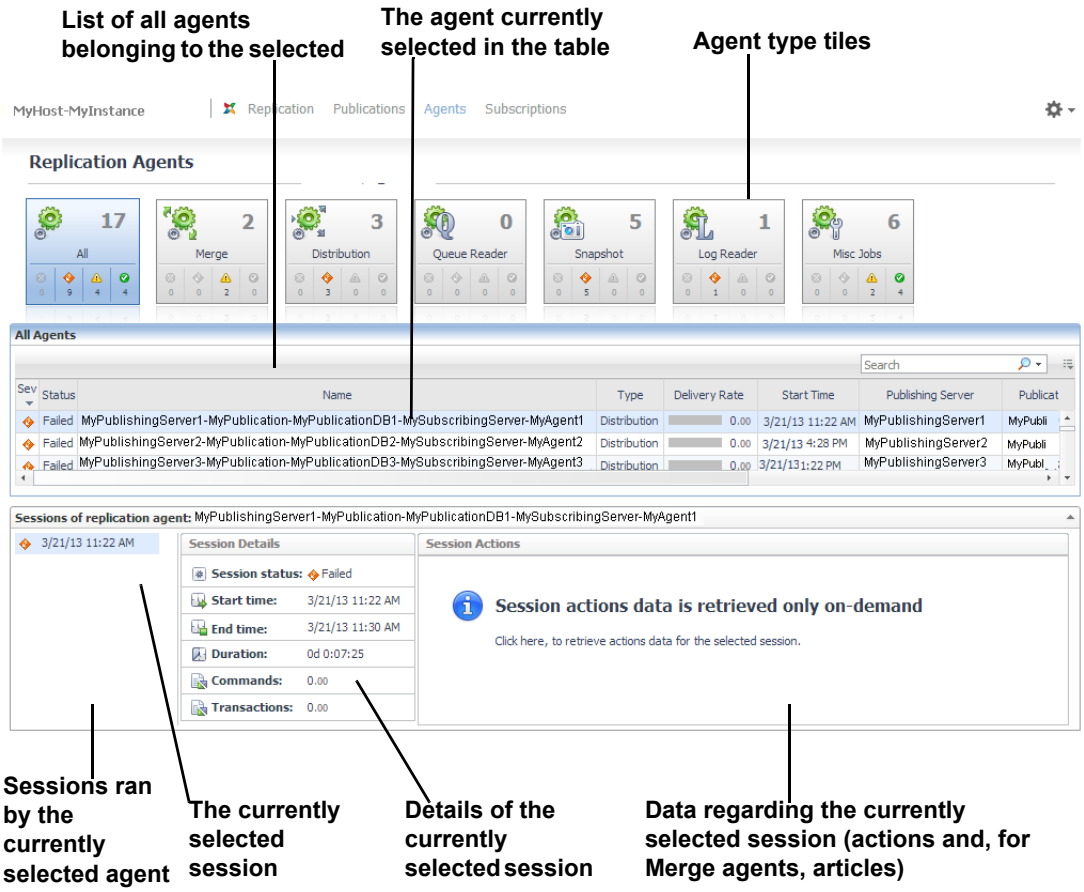
  - Subscription database
  - Subscription status — can have one of the following values: *Running*, *Failed*, *Retrying*, *Idle*, and *Not started*.

  IMPORTANT: To view a detailed representation of a specific subscription using the Replication Subscriptions drilldown, click the requested subscription's name. The Subscription Details screen opens, which provides detailed data about the selected subscription. For details, see The Subscription Details Screen on page 44.

Monitoring the Replication Agents

The Replication Agents drilldown provides detailed information of all the agents that were active during the selected time range. This drilldown allows you to filter the display to show agents of selected type or a selected severity level within a certain agent type, and review all the agent's session actions.

Figure 9. The Replication Agents drilldown.



This drilldown contains the following sections:

- Agent type tiles — used for configuring the list displayed on the *Agents* table; either displaying all agents by clicking **All**, or filtering the display by agent type or severity level.
- Agent table — provides detailed data regarding each of the agents or jobs selected to be displayed. For details, see Reviewing agent-specific data on page 41.
- Sessions ran by the currently selected agent — displays all the sessions run by the agent selected in the table during the specified time range. Selecting a session on the list displays its details on the middle and right sections.
- Details of the currently selected session — displays prominent statistics about the session currently selected on the list.
- Agent-specific session actions data — displays actions (and merge articles, for Merge agents) for all sessions run during the specified time range by the agent selected in the Agent table. For details, see Reviewing session activity data for the selected agent on page 42.

Reviewing agent-specific data

The Agents table displays all agents whose type tile was selected on the toolbar. Each row on this table provides the following data about a specific agent:

- Agent identification — the following agent-related data:
  - Highest severity level
  - Name
  - Type
  - Average delivery rate during the specified time range.

  NOTE: Hovering over this gauge displays a pop-up that contains a description of the flow, and a chart that shows the relevant data plotted over the specified time range.

  - Operation start time
- Entities involved in the publishing process — provides the names of the following components:
  - Publishing server
  - Publication database
  - Publication
  - Subscribing server
  - Subscription database

Except for agents of type Misc. Jobs, clicking each agent row displays the sessions ran by the selected agent during the specified time range. For details, see Reviewing session activity data for the selected agent on page 42.

Reviewing session activity data for the selected agent

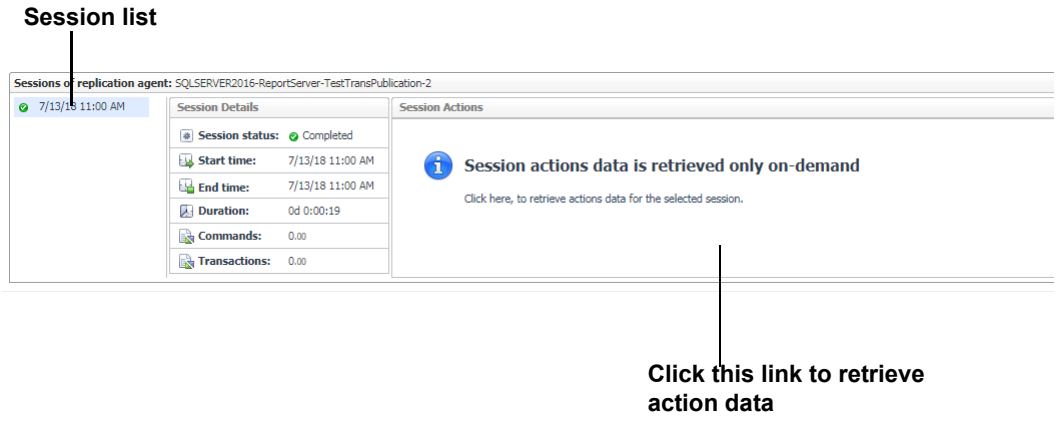
The lower section of the Replication Agents drilldown displays agent-specific session data for all agents whose type is other than Misc. Jobs.

To view the session actions of a single agent:

1 Click the requested agent's row on the Agents table.

Session data is retrieved only on demand by clicking a button or link, or by refreshing a screen component. The initial display of the Session section provides no data on the Sessions Actions pane.

Figure 10. Click this link to retrieve action data.



NOTE: When the selected agent's type is Merge, the Sessions section also displays the Session articles.

2 To retrieve action data for the first session on the session list, click the link under the message title.

To retrieve action data for any other session, click the requested session on the session list on the left.

The retrieved data is displayed on a detailed list of all actions performed by the session during the specified time range, indicating the time the action took place, and the action message.

Monitoring the Replication Subscriptions

The Replication Subscriptions drilldown allows performing an in-depth analysis of all subscriptions servers by the selected distributing instance.

To investigate the requested subscription:

1 Select which subscription type to display, by selecting the tile that represents the requested type: Snapshot, Transactional, Transactional with update, Merge or All.

> **IMPORTANT:** In each of these tiles, it is possible to filter the display to show only subscriptions with a specific severity status, by clicking the icon that represents that status. For example, Not Started.

2 Review the requested publication data on the *Subscriptions* table:

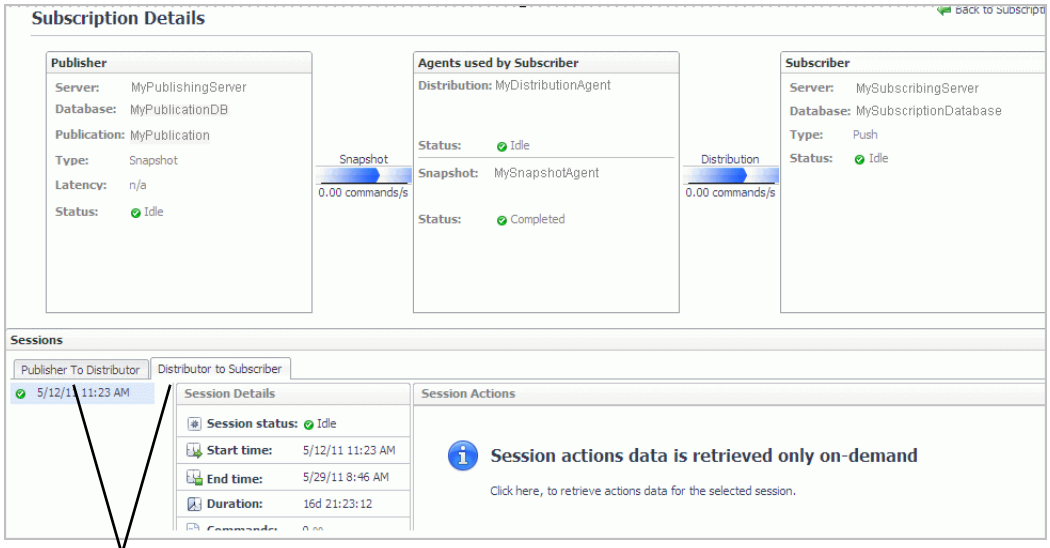
- Severity — the highest (worst) severity level
- Status
- Subscribing server name
- Subscription database name
- Subscription type — Pull, Push, or Anonymous
- Publication name
- Publishing server name
- Publication database name
- Publication type
- Agents taking part in the transaction — Log Reader, Queue Reader, and Snapshot agents

Click the subscription's row to access the **Subscription Details** screen.

The Subscription Details Screen

The Subscription Details screen allows tracking the process of replicating data to the selected subscription, by displaying all components involved in the process and providing links for further investigation.

Figure 11. The Subscription Details screen.



Data flow panes — from the Publishing server down to the Subscribing server (can be toggled)

Similar to the Replication home page, this screen contains the following panes, each representing a stage at the replication process:

- Publisher — provides the publication-related data that appears on the Replication home page, that is, publishing server, publication database, and publication name, and the following data:
  - Type
  - Latency
  - Status — can have one of the following values: *Running*, *Failed*, *Retrying*, *Idle*, and *Not started*.
- Agents used by Publisher — displays the replication agents used for replicating the publication data to the subscribing server.

  The agent types displayed on this pane for each publication type are as follows:
  - In *snapshot*, *Transactional* and *Merge* publications — Snapshot and Log Reader.
  - In *Transactional with update* publications — Snapshot, Log Reader, and Queue Reader.

  IMPORTANT: To carry out an in-depth analysis of the agent's actions using the Replication Agents drilldown, click the requested agent's row. The Replication Agents drilldown appears, with the clicked agent's row selected in the All Agents table. For details, see Monitoring the Replication Agents on page 41.
- Subscriber — provides the following data regarding each of the publication's subscriptions:
  - Subscribing server
  - Subscription database
  - Subscription type — Push, Pull, or Anonymous. For further details, see glossary definitions for Pull subscription on page 302, Push subscription on page 302, and Anonymous subscription on page 283.
  - Subscription status — can have one of the following values: *Running*, *Failed*, *Retrying*, *Idle*, and *Not started*.

- Sessions — provides agent-specific session data for all agents taking part in the selected subscription process. If the process involves more than one agent, this section is divided into the following panes:
  - **Publisher to Distributor** — displays the session actions of the Snapshot or Log Reader agent.
  - **Distributor to Subscriber** — displays the Distribution agent's session actions.

Monitoring Alarms

The *Alarms* pane contains a table that displays only replication-related alarms, invoked if the metrics defined in the replication-related rules have been exceeded during the specified time range (by default: the last 60 minutes). For a detailed list of such alarms, see Replication-related alarms on page 45.

Unlike the Alarms pane on the Foglight for SQL Server home page, which always displays all alarm types, this table allows alternating between two display methods:

- Displaying all alarms — by clicking **All**.
- Filtering the alarm display by severity level — by clicking a specific severity level.

IMPORTANT: Fatal alarms are invoked when the SQL Server agent stops responding, and are not due to replication errors. Therefore, the count of alarms whose severity level is Fatal (indicated in red) should always remain 0. The alarm display should be filtered by clicking either the Warning or Critical severity level.

NOTE: To view detailed information regarding each alarm, go to the **Replication Agents** drilldown.

Figure 12. Alarm Indicator Descriptions

Indicator	Description
Yellow (Warning)	The metric (or workload resource) has exceeded the configured threshold or deviated from the baseline.
Orange (Critical)	The metric (or workload resource) has exceeded a threshold or extremely deviated from the baseline.
Red (Fatal)	The SQL Server is not responding, triggering a fatal alarm.

Replication-related alarms

The replication-related alarms are as follows:

- Replication Agents - Delivery Latency — invoked when the delivery latency for a specific instance exceeds a predefined threshold.
- Replication Agents - Status — invoked when the replication agent fails to start.
- Replication Publications Due to Expire — invoked when the number of hours left until the replication publications are due to expire falls below a predefined threshold.
- Replication Publications - Errors — invoked when an error is encountered in one of the replication publications.
- Replication Publications - Status — invoked when a replication publication fails to start.
- Replication Subscriptions - Status — invoked when a replication subscription fails to start.
- Replication Available — invoked when the replication process has encountered a failure.

For details, see the *Foglight for SQL Server Reference Guide*.

Monitoring SQL Performance

The SQL Performance page provides the ability to investigate the activity and resource consumption of a selected instance. There are two levels of metrics available:

- Without SQL PI configured — basic History, lock analysis data, and activity highlights. See Monitoring SQL Performance without SQL PI configured on page 50
- With SQL PI configured — The ability to perform a more in-depth analysis and investigation of the Instance activity and resource consumption by adding a dimension view of the activity, change tracking analysis, execution plan analysis and compare toll. See Monitoring SQL Performance with SQL PI Configured on page 53.

The SQL Performance page displays the following components:

- The **Instance View** — Provides the ability to investigate and analyze the resource consumption of the instance.
- The **Resource toolbar** — By selecting one of the resources, the resource consumptions charts and the overview section are updated to display only the relevant information regarding the activity of the selected resource.

Resource Toolbar Options

To review specific data metrics, select one of the specific metrics from the toolbar. The table below lists which metrics are available under each toolbar option.

Table 6. Metrics Available under each Resource Toolbar option.

Resource Toolbar Option	Default Metrics Available
Workload Panel	Allows you to review the instance workload by gathering all resources into one view. Default metrics displayed: • Blocked Lock Requests • Latch Wait • Total CPU Usage • Procedure Cache Hit Ratio • Virtual Memory Used • Availability
CPU	The CPU panel allows you to review the Instance CPU usage and CPU wait events. Default metrics displayed: • Total CPU Usage • CPU Wait

Table 6. Metrics Available under each Resource Toolbar option.

Resource Toolbar Option	Default Metrics Available
I/O	The I/O panel allows you to review the I/O-related data, such as wait events and physical reads and writes. Default metrics displayed: • I/O Wait • I/O Bulk Load Wait • I/O Completion Wait • I/O Data Page Wait • Lazy Writes • Checkpoint Pages • Page Splits • Page Life Expectancy • Disk Utilization
Memory	The Memory panel allows you to view memory related performance data, such as wait events and physical reads and writes. Default metrics displayed: • Page Life Expectancy • SQL Server Cache memory • SQL Server Connections Memory • Target Instance Memory • Total Instance Memory • Memory Wait
Network	The Network panel displays the network related wait events. Default metrics displayed: • Network Wait • Network HTTP Wait • Network I/O Wait • Network IPC Wait • Network Mirror Wait
Lock	The Lock panel displays the database's lock related wait events. Default metrics displayed: • Blocked Lock Requests • Table Lock Escalation • Lock Requests • Lock Wait • Lock Bulk Update Wait • Lock Exclusive Wait • Lock Intent Wait • Lock Schema Wait • Lock Shared Wait • Lock Update Wait
Latch	The Latch panel displays the database's latch related waited events. Default metrics displayed: • Latch Wait • Latch Savepoint Wait

Table 6. Metrics Available under each Resource Toolbar option.

Resource Toolbar Option	Default Metrics Available
Log	The Log panel displays the database's log related wait events. Default metrics displayed: • Log Flushes • Log Wait • Log Buffer Wait • Log Other Wait • Log Synchronization Wait • Log Write Wait
CLR	The CLR panel displays the database's wait events occurring as a result of statements waiting for CLR code execution to complete. Default metrics displayed: • Total CPU Usage • CLR Wait
Remote Provider	The Remote Provider panel displays the databases wait events that take place when various processes are waiting either for a remote OLEDB call to complete or for DTS synchronization. Default metrics displayed: • Remote Provider Wait • OLEDB Provider Full Text Wait

Table 6. Metrics Available under each Resource Toolbar option.

Resource Toolbar Option	Default Metrics Available
XTP	The XTP (Extreme Transaction Processing) panel displays information gathered by the SQL Performance Investigator and XTP-related DMVs. Default metrics displayed: • XTP Transaction Wait • XTP Wait • XTP Miscellaneous Wait • XTP Log write Wait • STP Procedure Wait • XTP Transactions • XTP Failed due to Unique Constraint • XTP Writes • XTP Failed due to Write Conflicts • XTP Tranactions on Durable Tables • XTP Failed Due to Dependencies • XTP Read only Transactions • XTP Failed due to Validation • XTP Aborted Transactions
Other	The Other panel allows monitoring the time spent waiting for the completion of miscellaneous operations. That is, operations that cannot be classified into any other wait categories. Default metrics displayed: • Other Wait • Parallel Coordination Wait • Cursor Synchronization Wait • Backup Recovery Wait • Database Replication Wait • Deferred Task Worker Wait • External Procedure Wait • Full Text Search Wait • Hosted Components Wait • Other Miscellaneous Wait • Service Broker Wait • Synchronous Task Wait

In addition, the following non-default metrics can be added to each of the above resources:

- Active Time
- Active Transactions Total
- Disk Queue Length
- Full Scans
- Index Searches
- Instance Unavailable
- Instance Up Time
- MSSQL Physical I/O Operations
- Physical Page Reads
- Physical Page Writes

- Probe Scans
- Range Scans

Monitoring SQL Performance without SQL PI configured

Viewing Metrics

The History section view is divided into two sections that are correlated to each other:

- Resource consumption charts — This section displays data in five different charts:
  - Workload chart — Displays the instance resource activity over the selected time frame by emphasizing the resources by colors.
  - Baseline chart — Displays the instance workload compared to the baseline over time.
  - Breakdown chart — Activity of the instance by second.
  - Resource Breakdown Pie chart — Displays the resource breakdown usage by % of the total instance activity.
  - All wait events pop up — Displays details of the wait events that the instance is waiting on during the selected time range.
- Overview section- Displays a graphical representation of the metrics highlighted in the Workload related Metrics table below.
  - Workload related Metrics - A table that displays a variety of resource consumption metrics which can give an in-depth of the instance activity, each resource holds its default metrics.

NOTE: By using the metric selector other metrics can be added to the table.

Blocking History

The Lock Analysis displays all locks that took place within the selected time range.

The lock analysis feature is integrated as part of the performance tree and it displays all the lock trees including further details for both the blocker and the blocked session including:

- Lock event start date
- Session Identifier - [Sid,SERIAL#]
- Session Identifer of the blocker
- Locked Object name
- Status
- Lock Duration
- Program
- DB User
- SQL Text
- Client Machine

Viewing SQL Statement Details

The SQL Statements table displays the top SQL statements that experienced the longest elapsed time during the selected time range. The number of SQL statements displayed is shown at the table's title, together with the field used for ordering the SQL statements. Selecting a row displays the workload and executions of the selected SQL statement on the SQL Activity section below. For details, see SQL Activity section on page 51.

Selecting the SQL Text in the SQL Statements tree panel displays a page that allows viewing detailed information about the selected SQL statement. For details, see SQL Activity section on page 51.

NOTE: SQL Statement settings can be changed through **Administration > Top SQL Statements.**

To view SQL statement details:

1 In the Instance view, expand the Instance View.

2 Click SQL Statements.

Table 7. The Top SQL Statements table contains the following columns:

Column	Description
SQL Text	The SQL text for the selected SQL statement.
Object Name	The name of the object owning the SQL. For example: view, procedure or function.
% CPU Workload	The relative share, in percent, which the selected SQL statement occupies within the instance's total CPU consumption.
Executions Started	The number of the selected SQL statement's executions that started during the selected time range.
SQL % Hit Rate	The hit rate, in percent, of the selected SQL statement.
Elapsed Time	The total time consumed for carrying out the SQL statement executions.
CPU Time	The total CPU time consumed for carrying out the SQL statement executions.
Total I/O Operations	The total number of reads, both physical and logical, carried out by the selected SQL statement.

SQL Activity section

Located below the table of Top SQL Statements by Elapsed Time, the SQL Activity section of the pane displays the workload and executions of the SQL statement you selected in the TOP SQL Statements table.

These metrics are visually represented in two graphs:

- CPU Workload (Average Active Sessions) — Displays the activity (system workload), plotted over the specified time range. This graph compares the following metrics:
  - Overall CPU Workload — the workload incurred by the total top SQL statements displayed in the Top SQL Statements by Elapsed Time table.
  - Selected SQL CPU Workload — the average number of active sessions incurred by the selected SQL statement.
- Executions — Displays the number of times the selected SQL statement was executed during the selected time range. Alternatively, it is possible to display the number of executions carried out by the total SQL statements in the Top SQL statements table.

Viewing specific SQL statement details

To view detailed graphic and textual data about a specific SQL statement, select the SQL statement from the statements listed in the Instance View.

This page contains the following panes:

SQL activity

- The SQL Activity pane displays the CPU workload and executions of the selected SQL statement, as well as all SQL statements displayed on the Top SQL Statements table. This pane is identical to the SQL Activity section on the Top SQL Statements panel.

SQL text

- The SQL Text pane displays the short text of the selected SQL handle, and allows viewing the wider context of this text, namely: the SQL handle's full text, the entire batch and the SQL Statement's execution plan.
- This pane displays the short SQL statement, whose maximum size is by default 256 characters. In addition, the pane includes the following buttons:
  - View Full Text — Displays a pop-up with the SQL handle's full text
  - View Batch — Displays a pop-up with the full text of the entire batch to which the SQL handle belongs
  - View Plan — Displays a pop-up that shows the SQL statement's execution plan and allows downloading the execution plan (.sqlplan) file.

SQL metrics

Table 8. The SQL Metrics pane contains a table with the selected SQL statement's metrics:

Column	Description
Object Name	The name of the object from which the SQL statement was executed.
Executions	The number of executions that took place on this object since it was brought into the library cache. **NOTE:** The Executions parameter does not provide any informational value unless the Total option is selected. When selecting the option Perf Execution, the value of this parameter is always one (1).
Elapsed Time	The total time consumed for carrying out the SQL statement executions.
CPU Time	The total CPU time consumed for carrying out the SQL statement executions.
CLR Time	If the statement executed one or more CLR objects, the time consumed for these executions.
Total I/O Operations	The total number of reads, both physical and logical, carried out by the selected SQL statement.

Pie Charts

The Pie Charts pane includes the pie charts listed below, which display the activity during the selected time range for the selected metrics.

- Total CPU Time — The total CPU time consumed for executing the selected SQL statement, compared with the CPU time consumed for executing all of the SQL statements
- Executions — The number of times the SQL script executed for a the selected SQL handle, compared with the total SQL statements
- Logical Reads — The total number of logical reads for the selected SQL statement, compared with the total SQL statements
- Physical Reads — The total number of physical reads for the selected SQL statement, compared with the total SQL statements

Monitoring SQL Performance with SQL PI Configured

The SQL Performance Investigator provides the ability of a more in-depth analysis and investigation of the Instance activity and resource consumption.

SQL PI provides the ability to investigate and analyze the resource consumption of the instance by using:

- Activity Highlights. See Activity Highlights on page 53
- Performance tree. See Performance Tree on page 53
- History. See Viewing Historical Metrics on page 54
- Change Tracking. See Viewing Change Tracking on page 59
- Execution plan. See Viewing Execution Plans on page 60

Activity Highlights

The Activity Highlights are provided for fast performance analysis and allow users focus on the most significant dimensions that are relevant for the resource selected in the selected time range.

This pane comprises the following elements:

- Summary — Summarizes the instance consumption time by % of the total instance activity.
- Activity Highlights table — Displays the activity highlights, resource breakdown usage, and the top wait event that the instance is waiting on within the selected time range.

Performance Tree

The performance tree provides iterative (up to three levels) access to any of the key dimensions associated with SQL Server database activity, based on the OLAP multidimensional model and an instance view of the instance activity. Domain nodes offer a hierarchical view of all types of SQL Server activity characteristics.

Selecting a dimension from the tree determines what subset of activity is displayed. Iterative drill-down into domains of interest provides increasingly refined focus and diagnosis.

For example, to begin the investigation by first identifying the most active DB User, follow the steps described below:

1. Select the DB Users node, to display the most active database users in the selected time range. That is, the database users who consumed the highest amount of the selected resource.
2. Select the first user, to focus the entire window on that user's activity.
3. Identify the most demanding SQL statement that this specific user has executed, by expanding the user node and then selecting the SQL statement dimension node.

   This displays the most active SQL statements executed by this user.
4. Select a specific SQL statement to focus the entire window on the selected statement's activity.
5. Select **Client Machines** under the selected SQL Statement, to view the computers on which the statement was run.

In a similar manner, such iterative drilldowns can be carried out into any SQL Server dimension of interest, to gain a complete understanding of the causes of its behavior.

The default SQL Server dimensions are as follows:

- SQL Statements — The executed SQL commands.

- TSQL Batches — T-SQL Batch is the set of T-SQL (Transact-SQL) commands that are sent to execution together, usually corresponding to a single business transaction. TSQL Batch can end with a GO command, and execute database components such as:
  - Stored Procedures — Sets of T-SQL code that are stored and compiled in a SQL Server database.
  - Functions — Saved T-SQL routines that return a single value or a set of columns and rows.
  - Triggers — Batches that fire after the execution of a DML (Data Manipulation Language) statement.
- Databases — The database context in which the session carried out its operations. A session may switch to numerous databases within its lifetime.
- Programs — Name of a program that connects to SQL Server and executes the SQL statements, as specified in the Program column of the SQL Server session information.
- OS Users — Operating system users running the client program.
- Client Machines — The machines on which the client executable (connected to SQL Server) is running.
- DB Users — SQL Server login names used for logging in to SQL Server.
- Context Info — Optional trace information that a session can create using the SET CONTEXT_INFO command. Context Info allows users to associate up to 128 bytes of binary information with the current session or connection. While the technical implementation is binary, users commonly associate keywords or strings of text with their sessions.
- Command Types — Executed SQL command type (for example, INSERT and SELECT).
- Sessions — Presents the top sessions which consumed the highest active time during the selected time frame.
- Locked Objects — Displays the objects that experienced locks, the duration of the lock and the type of the lock. The object view can be sorted by selecting a dimension in the performance tree, for example: by selecting a database name only the locked object that occurred on that database will be displayed.

In the SQL Statements and SQL Batches dimension there is an option to view information regarding the selected statement\batch through the top SQL statements\TSQL Batches:

- View Full Text — View full text of the statement\batch selected.
- Analyze Plan — View execution plan analytics of the statement\batch selected for more information regarding the execution plan. See Viewing Execution Plans on page 60
- Tune SQL — Export the statement\batch selected to a file that can be opened by Quest's SQL Optimizer toll.
- Compare — Retrieve the selected statement\batch to the compare dashboard for comparing it to other activities. For more information regarding the Compare, see Comparing Performance on page 61.

Viewing Historical Metrics

The History section view is divided into two sections that are correlated to each other:

- Resource consumption charts — This section displays data in five different charts:
  - Workload chart — Displays the instance resource activity over the selected time frame by emphasizing the resources by colors.
  - Baseline chart — Displays the instance workload compared to the baseline over time.
  - Breakdown chart — Activity of the instance by second.
  - Resource Breakdown Pie chart — Displays the resource breakdown usage by % of the total instance activity.
  - All wait events pop up — Displays details of the wait events that the instance is waiting on during the selected time range.
- Overview section- Displays a graphical representation of the metrics highlighted in the Workload related Metrics table below.

- Workload related Metrics - A table that displays a variety of resource consumption metrics which can give an in-depth of the instance activity, each resource holds its default metrics.

Selecting each dimension in the performance tree together with a specific resource effects the data displayed for each Level.

For example, by selecting the Lock resource the Instance view dimension will present only locks related data, the SQL Statements dimension will present only the statements that were experiencing locks and DB users the were experiencing locks and so on through all the dimensions and resources.

Resource Toolbar Options

Table 9. Resource Toolbar options and the instance metrics displayed.

Resource Toolbar Option	Instance Level Default Metrics	Dimension Type and Dimension Value Default Metrics
Workload Panel	The Workload resource is selected by default • Active Time • Executions • CPU Usage • Average SQQL Response Time • Wait Time Percent • Logins Rate • Batches Rate	• Active Time • Executions • CPU Usage • Average SQL Response Time • Wait Time Percent • Row Count
CPU	The CPU panel allows you to review the CPU usage and CPU wait events. • CPU Usage • CPU Wait • Host Run Queue Length • Host CPU Usage • Host Non SQL CPU utilization	• CPU Wait • CPU Usage • CPU Time • CPU Wait • CPU Usage
I/O	The I/O panel allows you to review the I/O-related data, such as wait events and physical reads and writes. Default metrics displayed: • I/O Completion • I/O Data Page • Latch Buffer • Host Disk Utilization • Logical Reads • Deferred Task Worker • Physical Reads • I/O Wait • Writes • Host Disk Queue Length • I/O Bulk Load	• I/O Completion • I/O Data Page • Latch Buffer • Logical Reads • Deferred Task Worker • Physical Reads • I/O Wait • Writes • I/O Bulk Load

Table 9. Resource Toolbar options and the instance metrics displayed.

Resource Toolbar Option	Instance Level Default Metrics	Dimension Type and Dimension Value Default Metrics
Memory	The Memory panel allows you to view memory related performance data, such as wait events and physical reads and writes. Default metrics displayed: • Total Instance Memory • Pages out • Pages in • Buffer Cache Hit Ratio • Memory Growth Pressure • % Total Server Memory • Target Instance Memory • Page Life Expectancy • Memory Wait • Procedure Cache Hit Ratio • Host Physical Memory	• Cache Hit Ratio • Memory Wait
Network	The Network panel displays the network related wait events. Default metrics displayed: • Network IPC • Network Mirror • Network Wait • Packets received • Packets sent • Instance Packets Total • Network HTTP	• Network IPC • Network Wait • Network Mirror • Network I/O • Network HTTP
Lock	The Lock panel displays the database's lock related wait events. Default metrics displayed: • Lock Update • Lock Shared • Average Lock Duration • Lock Wait • Lock Intent • Lock Timeouts • Lock Exclusive • Deadlocks • Lock Schema • Lock Bulk Update	• Lock Update • Lock Shared • Lock Wait • Lock IntentLock Exclusive • Lock Bulk Update • Lock Schema
Latch	The Latch panel displays the database's latch related waited events. Default metrics displayed: • Latch Savepoint • Latch Wait • Internal Catch Latch	• Latch Savepoint • Latch Wait • Internal Catch Latch

Table 9. Resource Toolbar options and the instance metrics displayed.

Resource Toolbar Option	Instance Level Default Metrics	Dimension Type and Dimension Value Default Metrics
Log	The Log panel displays the database's log related wait events. Default metrics displayed: • Log Wait • Log Write • Log Buffer • Log Other • Log Synchronization	• Log Wait • Log Write • Log Buffer • Log Other • Log Synchronization
CLR	The CLR panel displays the database's wait events occurring as a result of statements waiting for CLR code execution to complete. Default metric displayed: • CLR Wait	• CLR Wait
Remote Provider	The Remote Provider panel displays the databases wait events that take place when various processes are waiting either for a remote OLEDB call to complete or for DTS synchronization. Default metrics displayed: • Remote Provider Wait • Distributed Transaction • OLEDB Provider Full Text	• Remote Provider Wait • Distributed Transaction • OLEDB Provider Full Text

Table 9. Resource Toolbar options and the instance metrics displayed.

Resource Toolbar Option	Instance Level Default Metrics	Dimension Type and Dimension Value Default Metrics
Other	The Other panel allows monitoring the time spent waiting for the completion of miscellaneous operations. That is, operations that cannot be classified into any other wait categories. Default metrics displayed: • Parallel Coordination Wait • Other Wait • Database Replication • Always On • Other Wait • Hosted Component • Synchronous Task • Service Broker • Other Miscellaneous • Full Text Search	
XTP	• XTP Wait • XTP Transaction Wait • XTP Procedure Wait • XTP Log write Wait • XTP Miscellaneous Wait • XTP Transactions • XTP Read Only Transactions • XTP Aborted Transactions • XTP Failed due to Validation • XTP Failed Due to Dependencies • XTP Failed due to Unique Constraint • XTP Failed due to Write Conflicts • XTP Writes • XTP Transactions on Durable Tables	• XTP Wait • XTP Transaction Wait • XTP Procedure Wait • XTP Log write Wait • XTP Miscellaneous Wait

Viewing XTP Data

The XTP pane displays information gathered by SQL Performance Investigator and XTP-related DVMVs. Metrics displayed are:

Table 10. XTP metric definitions

Metric	Description
XTP Transaction Wait	Time spent waiting for in memory transaction events.
XTP Wait	Time spent waiting for in memory OLTP (XTP) related events.
XTP Miscellaneous Wait	Time spent waiting for in memory miscenllaneous events. Those events are not related to transactions, log or natively compiled stored procedures.
XTP Log write Wait	Time spent waiting for in memory log and checkpoint related events.
XTP Procedure Wait	Time spent waiting for events related to natively compiled stored procedures.
XTP Transactions	The total number of transactions that have run in the In Memory OLTP database engine.

Table 10. XTP metric definitions

Metric	Description
XTP Failed due to Unique Contraint	Total number of unique constraint violations.
XTP Writes	Total number of megabytes written to the In Memory OLTP log records.
XTP Failed due to Write Conflicts	The number of conflicts between transactions while writing a row which lead to aborts.
XTP Transactions on Durable Tables	Total number of transactions that require log IO. Only considers transactions on durable tables.
XTP Failed due to Dependencies	The number of times a transaction aborts because a transaction on which it was dependent aborts.
XTP Read Only Transactions	The number of read only transactions.
XTP Failed due to Validation	The number of times a transaction has aborted due to a validation failure.
XTP Aborted Transactions	The total number of transactions that were aborted, either through user or system abort.

Monitoring and Tuning the Highest XTP Wait Events

In general, the XTP option on the SQL performance dashboard should show none or almost none XTP wait consumption. The most likely way to observe any XTP activity would be to show background processes.

However, in case there are significant waits related to XTP activity; use the Resource Breakdown pie chart. Clicking on the "All Wait Events" anchor allows you to get the actual wait counts. This drilldown display also allows you to compare the XTP wait events in comparison with other waits detected by SQL PI.

Investigation XTP Transaction Failures

Use the XTP related Metrics:

- xtp_aborted_transactions
- xtp_failed_due_to_validation
- xtp_failed_due_to_dependencies
- xtp_failed_due_to_unique_constraint
- xtp_failed_due_to_write_conflicts

These metrics allow you to find if there were XTP transactions that failed. They also enable research on the top reasons of failure. Clicking on each of these values allows you to investigate the amount of failure over time.

Viewing Change Tracking

The change tracking tool is an integrated monitoring mechanism. It periodically tracks changes in environments and activity that can potentially influence system performance and enables the user to view correlation between occurrences of changes and SQL Server's activity and behavior patterns.

Use the Categories filter mechanism to refine the set of displayed change tracking occurrences. These categories are displayed on the right hand side of the pane and include:

Accounts

Addition and deletion of login accounts.

Database Configuration

For each database in an MSSQL instance: configuration parameters, File group parameters; files (data and log) location, size and status.

Database Objects

For each database in an MSSQL instance: Any schema object: Tables and indexes and their partitions and sub-partitions; clustered and non-clustered indexes; constraints; views; functions; procedures; types and triggers (for example, altering a table; altering or rebuilding an index).

Execution Plan

SQL PI captures SQL statements whose execution plans have changed. When an execution plan has (unintentionally) changed, the outcome can result in SQL performance degradation. Therefore it is essential you investigate the change, by clicking the Properties button to graphically display the execution plan before and after the change, to verify that the nature and outcome of the change improve SQL performance.

Master Configuration

Track changes to sp_configure settings, and DBCC trace flags.

System Configuration

Hardware and operating system configuration (for example, disks and network interfaces (NIC); amount of RAM; CPU count; device installations and swap space allocation).

NOTE: Selecting a specific dimension from the Performance Tree will display only the changes made in this specific dimension. The instance view displays all changes made.

Viewing Execution Plans

This view presents the execution plan of a selected SQL\batch and the cost of it. A Historical execution plan can be generated and will present any two views of the execution plan.

There are two ways to access the Execution Plan History dialog:

- From a Change Tracking pane by clicking on a row that displays Execution Plan change.
- From History by selecting the statement or batch and by clicking the 'Analyze' button in the top of the table.

The execution plan dashboard displays the following content.

Top Bar

Date — Displays all the execution date and time of the selected statement\batch

Type — Displays the type of the execution plan.

Plan handle — Displays the SQL handle on the execution plan.

Plan analysis section

The execution plan is displayed in tabs which present the cost of each operation and object of each step of the plan.

By pressing the Generate Plan button the execution plan is generated and can be viewed by clicking on **Compare Plan**. This opens a pop up with a compression of all available execution plans of the selected statement.

Blocking History

Provides the same view as without SQL PI configured. See Blocking History on page 50.

Activity Highlights

Provides the same view as without SQL PI configured. See Viewing SQL Statement Details on page 51.

Comparing Performance

Use the SQL PI Compare where differences in period activity illustrate underlying performance and monitoring issues. It helps you determine whether a comparison occurrence is an isolated incident or a sign of a potentially significant performance problem.

Compare can be access from the Performance tree.

The comparison can be of whole instances or selected dimension breakdowns (such as user or SQL). Use Compare to address questions such as:

- What caused a specific activity?
- What were the resource and load demands of today's instance activity compared with that of a previous day?
- Is an instance imposing different load levels now than previously?
- How do we explain the difference in a SQL performance compared with a previous period?
- What are the differences in program characteristics over two periods that caused different performance?
- After identifying the different resource usage, you can use compare to identify what caused this difference: Usage pattern? Different SQL behavior? Environmental problem?

The upper panes graphically displays the Workload or any other selected resource.

The middle Activity section displays the dimensions which were significant elements of the difference. Expanding the individual lines displays the dimension members which caused the difference and the composing metrics whose differences exceed the specified threshold.

Use the set of performance related metrics (defined by the chosen resource) to help you explain the difference in resource consumption; for example, a rise in I/O Wait might be explained by a rise in the quantity of physical reads.

Use the Comparison Parameters section in order to enter all the desired information for the cooperation. For example, time range, instance, dimension, time frames, resource.

Reviewing Memory Usage

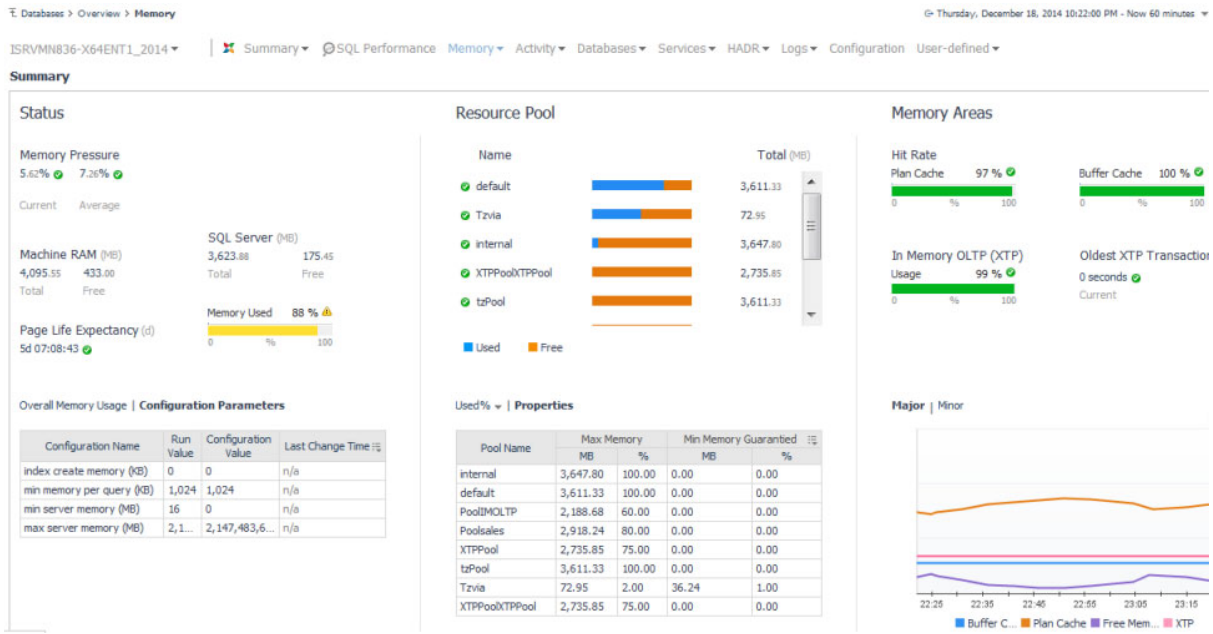
The Memory drilldown provides access to graphs that display details of memory usage for the currently monitored server.

Viewing the Memory Summary

The **Summary** panel allows monitoring how the SQL Server instance is using its physical memory. This panel features charts that allow you to perform a full investigation of memory-related issues on the server.

Use this panel to review any possible memory-related performance issues, and then go to the Buffer Cache or Plan Cache panels for an in-depth investigation of such issues.

Figure 13. The Summary panel



The Summary panel allows you to perform the tasks described in the following topics:

- Monitoring Status on page 62
- Investigating Resource Pool Memory Issues on page 63
- Investigating Cached Memory Issues on page 63

Monitoring Status

The Status chart displays the main areas used by SQL Server for carrying out database operations within memory. Use these charts to review the efficiency of the memory management, as reflected in the amount of memory allocated to each area.

Memory pressure is defined as the (100-100*Total Server Memory/Target Server Memory) or in other words, 1-accepted memory/requested memory by the SQL server. The percentage division shows the current memory pressure and the amount of memory used or free by SQL Server or entire server memory

Clicking on either of the **Machine RAM** or the **SQL Server** metrics allow you to investigate how these values changes over time.

Page Life expectancy is the length of time that a database page will stay in the buffer cache without references. A value too low for your system indicates that pages are being flushed from the buffer pool too quickly. The longer a page can stay in the buffer pool and be read from memory the better.

The **Memory Used** metric indicates the total percentage of dynamic memory, within the host's physical RAM, that the server is currently consuming.

Alerts regarding memory pressure may indicates that:

- More RAM Memory is required for the server.
- The processes running on the SQL server may consume more memory than possible for a healthy server.

The chart at the bottom of this section displays the metrics over time.

Selecting the Configuration Parameters link allows you to view the memory related options on the server.

Investigating Resource Pool Memory Issues

The Resource Pools (RP) metrics are supported only beginning with SQL Server v2008.

The middle division of the Memory Summary dashboard provides metrics which may help you to prevent resource pool memory related errors. For example:

- There is insufficient system memory in resource pool <RP Name> to run this query.
- There is insufficient memory available in the buffer pool.

The first chart in the Resource Pool section displays the resource pools available for application and server usage and alerts memory utilization if occurs on any resource pool. The values are grayed out when the resource governor is disabled.

The list contains the following values:

- An alert icon for the RP utilization.
- The name of the RP.
- A bar showing the usage within the RP.
- The total memory potentially available for the RP.

Two resource pools "internal" and "default" are created when SQL Server is installed. Clicking on the resource pool displays how the usages changes over time.

Further drilling into the databases related to this resource pool is also available as of SQL Server v2014, for databases which have the In-Memory feature enabled. When using In-Memory databases, it is highly recommended to define a dedicated resource pool.

The lower part of the Resource Pool section provides metrics on up to eight resource pools over time. Click the Properties tab to view the definition parameters of the resource pools on the server

Investigating Cached Memory Issues

The Memory Plan metrics provide data on cache-related issues.

Slow running processes and Hit ratio issues

Plan cache is a component of SQL memory that stores query plans for re-use. When a query is issued against SQL, the optimizer attempts to re-use a cached plan if the traits of the query permit - but can only do so if that plan resides in cache, otherwise it needs to compile a new plan.

Low hit ratio may indicate a lack of memory resources, a new plan might also be compiled even if a similar plan exists in cache when:

- There are changes in the schema.
- A query is running parallel that may have run serially before.

- Parameters have changed.

Plan compilations are expensive though; if plans are flushed from cache only to be recompiled afterwards because of size constraints even more memory resources are consumed.

Selecting the Plan Cache hit ratio allows you to investigate changes over time. Further drilling down leads to the Plan Cache dashboard.

Buffer cache (Also called Buffer Pool) is the place in system memory that holds data and index pages read from disk. This has two purposes:

- To modify pages according to INSERT and UPDATE statements. Those pages are marked as "dirty" and are flushed to disk when a checkpoint is performed.
- To increase response time when retrieving the same data.

Buffer Cache size is determined among other things by server memory and the target server memory specified in the Max Server Memory parameter. When that threshold is reached and SQL Server needs to read more pages, previously cached pages are discarded.

A low hit ratio may indicate lack of memory. It also indicates that the Buffer Cache cannot improve performance by reducing IO operations. As such, the Buffer Cache size can correlate to improved performance during a heavy workload.

Improving Buffer Cache hit ration in SQL Server 2014 is also possible by setting a Buffer Pool Extension on an SSD drive.

The hit rate for this cache should normally be above 90%. Selecting the Buffer cache hit ratio bar allows you to investigate these metrics over time. Further drilling down leads to the Buffer Cache dashboard.

Investigating In-Memory OLTP (XTP) Performance

New in SQL Server 2014, In-Memory OLTP can significantly improve OLTP database application performance.

The **Memory Usage** of In-Memory (XTP) indicates the amount of data used by memory optimized tables. A large percentage indicates a server that heavily uses In-Memory OLTP features. It is recommended that you dedicate a specific resource pool to databases containing In-Memory OLTP objects hence ensuring proper allocation of memory resources among applications on the SQL Server.

The **Oldest XTP Transaction** currently active on the server indicates issues regarding long running XTP transactions. Any value over 20 seconds is regarded a warning to an un-healthy behavior on the server. Long-running transactions should be avoided with memory-optimized tables. Such transactions increase the likelihood of conflicts and subsequent transaction terminations. A long-running transaction also defers garbage collection.

The longer a transaction runs, the longer In-Memory OLTP keeps recently deleted row versions, which can decrease lookup performance for new transactions.

Selecting the Oldest XTP transaction displays general information regarding the transaction and the session that had initiated it.

Investigating Major and Minor Memory Areas

The **Major Memory Areas** chart displays the amount of memory allocated to the Buffer cache, Plan cache, free memory and XTP. Use this chart to ensure that:

- The buffer cache has appropriate size of memory, which is sufficient to hold the most frequently used data pages.
- The free memory value is not consistently close to zero, which may indicate use of inefficient queries or shortage of SQL Server cache memory.
- The plan cache has appropriate size of memory, which is sufficient to hold the most frequently used execution plans.
- The XTP memory has no significant peaks that might mean there are memory utilization issues during the day.

The **Minor Memory Areas** chart displays the main areas SQL Server uses for carrying out database operations within memory. The display is broken down into each of the main cache areas:

- Lock Area — memory allocated to keeping track of locks.
- Optimizer Code — a work area for the SQL Server optimizer. For details, see glossary definition of Optimizer.
- Sort, Hash, Index — memory used for each of these operations.
- User Connections — memory allocated to keeping track of each connection's attributes.

Use this chart to review the efficiency of the memory management, as reflected in the amount of memory allocated to each area.

Monitoring Buffer Cache-related Data

The Buffer Cache panel allows investigating the utilization of the instance's buffer cache, by reviewing this cache's top-consuming objects, the page allocation, and the buffer cache hit rate.

Reviewing the Top-consuming Objects of the Buffer Cache

The Buffer Cache table displays the objects that currently occupy the most space in the buffer cache (the Top N buffer cache objects).

To improve performance, Foglight for SQL Server limits the number of records that can be displayed in this table to 300 rows, but by default, a maximum of 20 records are displayed.

NOTE: To define global settings about data retrieval to the Buffer Cache table panel, use the Buffer Cache view in the Databases Administration dashboard. To access this view, click the **In-context actions** button at the upper right side of the screen and then select **Agent settings**. For details, see Setting Options for Displaying Data in the Buffer Cache on page 158.

Table 11. The Buffer Cache table contains these columns:

Column	Description
Database	The name of the database where the object resides.
Owner	The user name of the object's owner.
Table	The table name (if the object is a table).
Index	The index name (if the object is an index). If this column is left empty, the object is a heap.
Cached Size	The amount of cache used by the object.
Index ID	The index identifier (if the object is an index). This metric can have one of the following values: • 0 — indicates that the object is a heap and not an index. • 1 — indicates that the object is a clustered index. • > 1 — indicates that the object is a non-clustered index.
File Group	The file group where the object resides.
% of Object	The percentage of object in the cache.
% of Cache	The percentage of buffer cache used by the object.

Because the process of determining the largest objects in the cache is highly CPU-intensive, this table can take some time to populate.

Tracking the Page Allocations

The Page Allocations chart displays the amount of memory allocated to database pages, free pages, and stolen pages over the selected time range (by default, last 60 minutes).

High volume of database pages requested by SQL Server can sometimes indicate that data is not properly indexed or that the query optimizer is not using the most efficient index.

Free pages values that are consistently close to zero may indicate use of inefficient queries, or shortage of SQL Server cache memory. The value of the Free Pages metric is calculated as follows: <free pages> * 8K / 1024. For details, see Free pages on page 292.

Constantly high value of stolen pages (pages that were taken from the buffer cache to satisfy other memory requests) may indicate an overall system memory shortage. For details, see Stolen pages on page 306.

Monitoring the Buffer Cache Hit Rates

The Buffer Cache Hit Rates chart displays the buffer cache hit rate over the specified time range.

The buffer cache hit rate is the rate of logical reads being satisfied from the buffer cache, saving the need to carry out physical reads.

The buffer cache contains database, free, and stolen pages, and its size is calculated as follows: <pages> * 8 K / 1024.

The buffer cache hit rate should normally exceed 90%; hit rate values that are constantly lower than 90% may result from extensive data operations, due to inefficient query planning.

Monitoring Plan Cache-related Data

The **Plan Cache** panel displays information about the SQL Server's plan cache, which holds execution plans for stored procedures, triggers, ad hoc SQL, and so on. This panel can be used for tracking the plan cache hit rate, and, if the rate it too low, investigate which object plans are used frequently.

The various panes of the **Plan Cache** panel allow carrying out the tasks described in the following sections:

- Monitoring the Plan Cache on page 67
- Reviewing Statistics about Plan Cache Objects on page 68

Reviewing Object Types

SQL Server 2005 and later objects

In SQL Server 2005 and later versions, the following object types can appear in the chart:

- **Bound Trees** — normalized trees for views, rules, computed columns, and check constraints. For details, see glossary definition of Bound trees on page 285.
- **Extended Stored Procedures** — a SQL Server object that dynamically loads and runs a function within a dynamic-link library (DLL) in a manner similar to a stored procedure. For details, see glossary definition of Extended stored procedures on page 290.
- **Object Plans** — query plans generated by creating a stored procedure, function, or trigger. For details, see glossary definition of Object plans on page 298.
- **Replication Procedure Plans** — query plans of a replication system stored procedure. For details, see glossary definition of Replication procedure plans on page 303.
- **SQL Plans** — query plans corresponding to statements prepared using *sp_prepare*, *sp_cursorprepare*, or using auto-parameterization. For details, see glossary definition of SQL Plans on page 305.
- **Temporary Tables and Table Variables** — temporary tables are session-specific tables, that is, the tables are automatically dropped when the session is closed. Table variables, on the other hand, are created in the memory and exist there until the running of a single Transact-SQL (T-SQL) batch is completed. For details, see glossary definition of Temporary tables and table variables on page 306.

Monitoring the Plan Cache

The Plan Cache table displays objects that are currently stored in the plan cache.

The plan cache contains executable plans for Transact-SQL, including objects such as stored procedures, triggers, views and defaults, as well as ad hoc and prepared SQL.

To improve performance, Foglight for SQL Server limits the number of records that can be displayed in this table to 300 rows, but by default, a maximum of 20 records are displayed.

NOTE: To define global settings about data retrieval to the Plan Cache table, use the Plan Cache view in the Databases Administration dashboard. To access this view, click the **In-context actions** button at the upper right side of the screen and then select **Agent settings**. For details, see Setting Options for Displaying Data in the Plan Cache on page 158.

Foglight for SQL Server retrieves the largest plan cache entries first, so if any entries are not displayed, they are the smallest ones.

Table 12. The *Plan Cache* table contains the following columns:

Metric	Description
Database	The database where the object resides.
Owner	The object's owner.
Object	The object name.
Type	The object type. An object can have one of the following types: • Stored procedure • Prepared statement • Ad hoc query • Replication procedure • Trigger • View • Default • User table • System table • Check • Rule
Use Count	The number of times this cache object has been used since the plan was created.
Reference Count	The number of objects that reference this cache object.
Used Size	The amount of space in the plan cache, in megabytes, which is allocated to this object.
% Cache	The percentage of plan cache used by this plan.
SQL Query Text	The name of the procedure, or the first 128 bytes of the batch submitted.
Language	The language of the connection that created the cache object. Each connection can individually set an SQL Server language to be used for the connection. **NOTE:** Running identical plans that use several, incompatible language formats decreases the plan cache hit rate. To avoid language issues, ensure that all identical plans use the same language.
Date Format	The date format used by the connection that created the cache object. **NOTE:** Running identical plans that use several, incompatible date formats decreases the plan cache hit rate. To avoid date issues, ensure using the same date format for all applications.

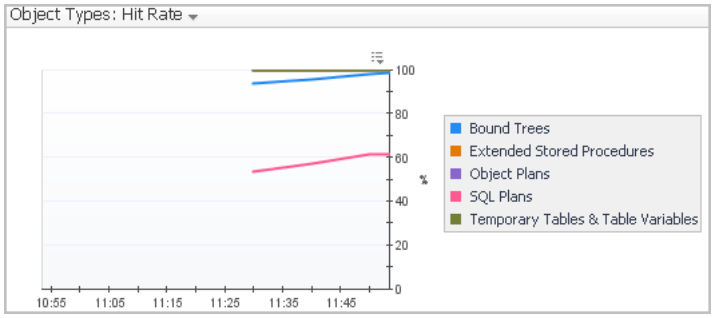
Tracking Memory Allocation to the Plan Cache by Object Type

The *Cache Size by Object Type* chart displays the amount of memory allocated to the plan cache over time, and breaks it down by object type. This display allows viewing trends of plan usage, which can be used for improving the queries used for retrieving the plans.

Reviewing Statistics about Plan Cache Objects

The *Object Types* chart shows various statistics for each type of plan cache object.

Figure 14. The Object Types chart.



The Object Types list can be used for selecting the metric to be displayed in this graph. Each metric is displayed for each type of plan cache object.

Table 13. The *Object Types* chart displays the following metrics.

Metric	Description
Hit Rate	The rate of logical reads that were satisfied from plans existing in the plan cache, divided by plan types.
Number of Objects	The number of objects of each type that are currently stored in the plan cache.
Use Rate	The rate at which each type of plan cache objects is being run (used).

Reviewing the Instance Activity

The Activity drilldown provides access to graphs that display both current and recent activity details for the currently monitored SQL Server instance.

The first section contains instructions regarding the use of the various panels of the Activity drilldown for carrying out a root-cause analysis of possible performance issues. The following sections provide in-depth data about each panel.

Viewing In-depth Data about the Instance

The Activity drilldown provides two levels of information about the currently diagnosed SQL Server instance:

- An overview of the activity in the currently monitored database, using the **SQL Instance Summary** panel.

  This panel can be used for indicating specific performance issues, such as high response time and low cache rates. Most of the possible reasons for a slow response time can be initially traced by using the charts in this panel. For details, see Reviewing the SQL Server Instance Activity on page 69.

- More detailed information, using the other panels.

  Each of the other panels allows carrying out a root-cause analysis of possible performance issues indicated in the SQL Instance Summary panel.

The Activity panels allow carrying out the following tasks:

- Viewing SQL I/O activity data — using the **SQL I/O Activity** panel, which provides graphical representations of various I/O activities of the currently diagnosed SQL Server instance. For details, see Viewing SQL I/O Activity Data on page 73.
- Viewing session data — using the **Sessions** panel, which lists all current SQL sessions and allows viewing session details and locks. For details, see Reviewing Session Details on page 86.
- Viewing detailed statistics about locks and latches — using the **Locks** panel, which displays information about all locks, latches and requests on the currently diagnosed SQL Server instance. For details, see Monitoring Locks and Latches on page 92.
- Viewing statistics about current lock conflicts — using the **Blocking (Current)** panel, which displays information about all current blocked sessions and lock conflicts, including the connections and resources involved in these conflicts. For details, see Tracking Current Lock Conflicts on page 93.
- Tracking deadlocks and their implications — using the **Deadlocks** panel, which displays all of the deadlocks that took place within the selected time range, as well as the databases and objects that were involved in the deadlock situations. For details, see Tracking Deadlocks and their Affected Objects on page 94.
- Viewing I/O Statistics by Database Files — using the **I/O by File** panel, which displays current I/O statistics for each SQL Server file. For details, see Viewing I/O Statistics by Database Files on page 97.
- Viewing the Real Activity data — represent the main activity areas in the SQL connection process. For details, see Viewing the Real-time Summary Page on page 73

Reviewing the SQL Server Instance Activity

The **SQL Instance Summary** panel displays a breakdown for the selected time range (by default: last 60 minutes) of system and SQL Server activity metrics, such as CPU utilization and response time. The display also includes the inner division within the metrics (for example, total CPU utilization compared with CPU utilization by the SQL Server). All metrics are displayed in the Foglight for SQL Server real-time summary; however, the home page displays only the last snapshot (by default: 20 seconds).

This panel allows viewing the source of a performance issue, by displaying the most immediate causes for this issue. Because the SQL Instance Summary panel displays the immediate performance indicators, most in-depth analyses can be carried out in other panels of the Activity drilldown.

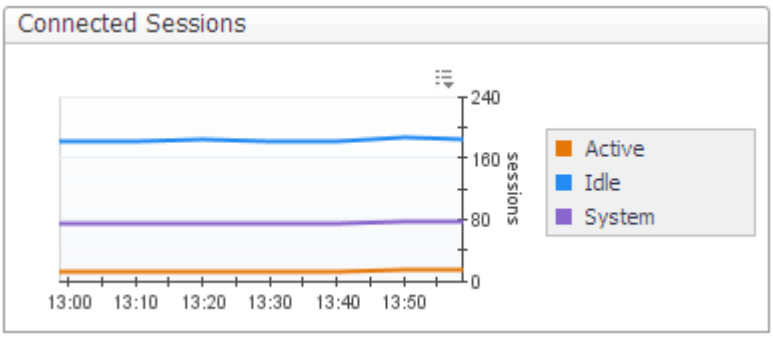
The **SQL Instance Summary** panel allows carrying out the tasks detailed in the following sections:

- Monitoring connected sessions on page 70
- Monitoring CPU utilization on page 70
- Viewing the SQL Server I/O activity on page 71
- Tracking the response time on page 72
- Monitoring cache hit rates on page 72
- Monitoring the call rates on page 73

Monitoring connected sessions

The *Connected Sessions* chart displays SQL Server session information. This chart contains a plot graph, which displays the number of SQL Server sessions over time.

Figure 15. The Connected Sessions chart.



Sessions are broken down into the following categories:

- Active user sessions
- Idle user sessions
- Internal SQL Server System sessions

A high number of active user sessions, that is, non-system sessions that are actively processing in SQL Server or that are waiting on locks, may indicate wait events that prevent such sessions from completing their activity and moving to idle state. For example, a CPU-intensive operation may result in a CPU wait event.

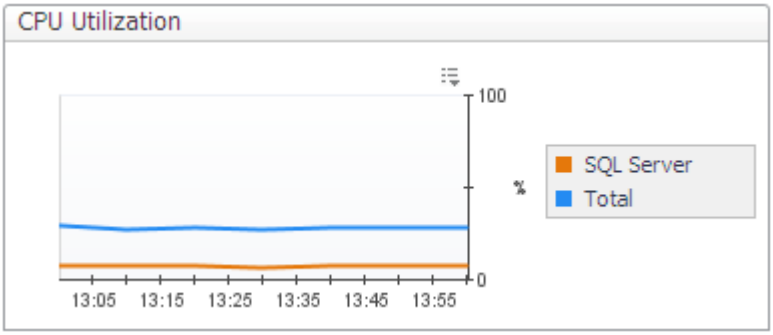
A high number of idle user sessions can complicate their management and lead to significant resource consumption, as a result of the update and management operations required for such sessions.

For more in-depth information about the SQL Server's sessions, go to the **Sessions** panel, described in section Viewing Session Details on page 84.

Monitoring CPU utilization

The *CPU Utilization* chart displays the amount of CPU being used by SQL Server compared with the total being used by all processes in Windows.

Figure 16. The CPU Utilization chart.



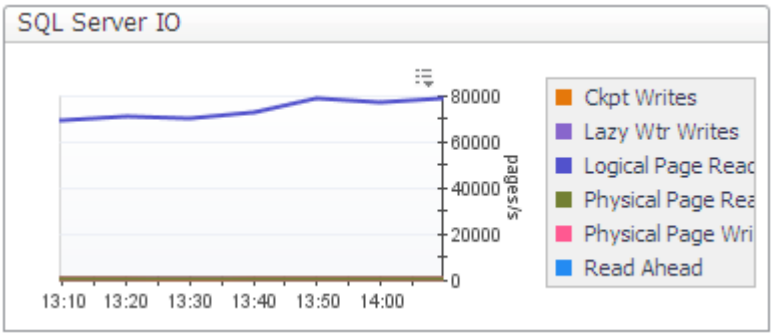
The top-consuming sessions in the system can be reviewed in the Sessions panel, by sorting the Sessions table by the CPU Usage column. For details, see Viewing Session Details on page 84.

Viewing the SQL Server I/O activity

The *SQL Server IO* chart displays the following types of I/O activity:

- Physical I/O activity — indicated by the rate at which pages are physically read from and written to disk by SQL Server.
- Logical I/O activity — indicated by the rate at which pages in the buffer cache, a memory area used by SQL Server to hold recently accessed database pages, are being referenced by SQL connections (logical page reads).

Figure 17. The SQL Server IO chart.



Under optimal work conditions, the SQL Server uses logical page reads to read pages from the buffer cache. However, required pages that do not yet reside in the cache are being read from disk using physical I/O operations.

A high value for the logical page reads indicates that SQL Server efficiently uses the memory allocated to its buffer cache. A high value for physical page reads, on the other hand, indicates that SQL Server is finding fewer pages already in memory, resulting in the need to perform more disk reads.

The SQL Server IO chart, which displays I/O activity for all sessions and for all SQL Server database files, is actually a summary of two different charts, found in the SQL I/O Activity panel. For details, see Monitoring the SQL Server physical I/O activity on page 82 and Monitoring the SQL Server logical I/O activity on page 83. A more focused display can be obtained using the steps detailed below.

To view a breakdown of logical and physical reads per session:

1. Click **Activity > Sessions** to go to the *Sessions* panel.
2. Click the requested session in the *Sessions* table.
3. Review the details displayed in the **Session Details** pane.

For details, see Viewing Session Details on page 84.

To view current SQL Server I/O statistics for each SQL Server database file:

1 Click **Activity > IO by File** to go to the *IO by File* panel.

2 View the details displayed in the *IO by File* table.

For details, see Viewing I/O Statistics by Database Files on page 97.

Tracking the response time

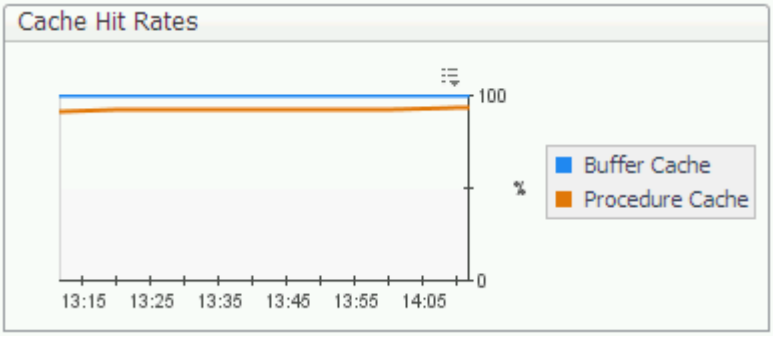
Response time is the full time (in milliseconds) it has taken a dummy query (by default: select 1) to get from the application to SQL Server and back.

Every time a real-time sampling interval starts (by default: 5 minutes), a query is sent and its response time value is displayed. Any value higher than 20 ms may indicate a performance issue.

The *Response Time* chart displays the response over the defined time range.

Monitoring cache hit rates

Figure 18. The *Cache Hit Rates* chart displays the hit rates for the main SQL Server cache resources.



A hit rate indicates the rate at which SQL Server finds pages already in the cache memory, saving the need to carry out physical reads.

Hit rates are shown for the following cache types:

- *Buffer Cache* — the hit rate for this cache should normally be over 90%.
- *Procedure Cache* — the hit rate for this cache varies widely, depending on how well the application is written.

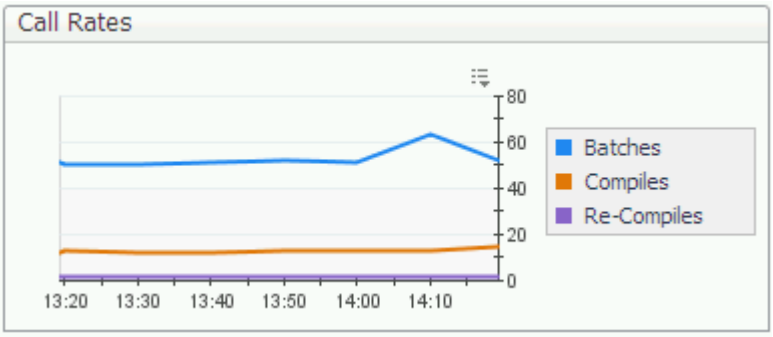
A buffer cache hit ratio value that continually goes below 90% may indicate the need for reviewing the memory settings.

Low procedure cache hit rates may indicate inefficient use of SQL cache memory, which increases the need for compilation.

To further investigate the reasons for the low cache hit rates, see Monitoring Buffer Cache-related Data on page 65 and Monitoring Plan Cache-related Data on page 66. To review how the cache hit rate affects the number of compiles and recompiles, review the Call Rates chart.

Monitoring the call rates

Figure 19. The *Call Rates* chart displays the rate at which various events are occurring in SQL Server.



The following metrics are charted:

- Batches — a batch is one or more transact SQL statements sent at one time from an application to the SQL Server instance. High rates of both batch, and compiles and re-compiles can sometime indicate a bottleneck, as a result of batches that are not using compiled plans, or identical batches that use a different plan or create a new one. To troubleshoot this issue, review the use of execution plans, date and language formats used.
- Compiles — the rate at which an SQL stored procedure is being compiled into the procedure cache. The Compiles figure includes Re-Compiles.
- Re-Compiles — the rate at which SQL Server is recompiling an SQL stored procedure in the procedure cache.

When many recompiles take place, the SQL Server's CPU can become overloaded, slowing down everything running on that computer. When a predefined threshold is exceeded, the Recompiles alarm is invoked. The Re-Compiles metric of this chart allows viewing whether the recompiling issue is a persistent one. For further details, see Recompiles Alarm on page 167.

Viewing SQL I/O Activity Data

The **SQL I/O Activity** panel provides information, represented graphically by charts, about the physical and logical activity of all SQL Server's sessions. In addition, this panel includes a chart that displays various statistics relating to how SQL Server data is being accessed and updated (for example, page splits and page allocations). An extra chart featured in this panel, Disk Queue Length, displays the disk activity for each logical disk, including activity generated by non-SQL Server processes.

The charts displayed in this panel provide a summary of physical and logical I/O activity for all sessions. To view a breakdown of these activities per session, go to the **Sessions** panel and click the requested session.

The **SQL I/O Activity** panel allows carrying out the tasks described in the following topics:

- Viewing the Real-time Summary Page on page 73
- Monitoring the SQL Server physical I/O activity on page 82
- Monitoring the SQL Server logical I/O activity on page 83
- Viewing how SQL Server's logical data is accessed and updated on page 83
- Viewing Session Details on page 84

Viewing the Real-time Summary Page

The following table identifies each of the main elements of the Foglight for SQL Server real-time summary page, and provides a link to display more information, corresponding to the logical dataflow within the Foglight for SQL Server Instance Homepage.

Table 14. Main elements of the Real-time Summary Page.

View Name	Description
Instance properties	Identifying the Instance on page 74
Sessions pane	Monitoring General Session Statistics on page 77
Background Processes pane	Monitoring Background Processes on page 81
Workload	The workload (average active sessions) graph for the SQL Server instance, plotted over the specified time range (by default: last 60 minutes).
Process activity	Data flows.Monitoring total activity on page 77
Memory Activity pane	Monitoring the SQL memory management on page 79
Disk Storage pane	Monitoring the disk storage on page 80
Physical I/O operations	Tracking physical I/O activity on page 80

The main groups (panes) of gathered icons and gauges, which represent the main activity areas in the SQL connection process, are as follows:

- Instance identification — used for identifying the instance, its type, and its properties. See Identifying the Instance on page 74.
- Components representing instance data flow — the main activity area in the Foglight for SQL Server Instance Homepage includes several panes and flows that represent the data flow in the SQL Server operation. The dataflow is described in this guide as a top-down design, that is: from the session to the physical disk storage. See Tracking the Instance Data Flow on page 76.

Overview Dashboard

Identifying the Instance

The Instance identification indicators allow you to identify the currently monitored instance, its type and its operation period.

These indicators are as follows:

- Instance name — identifies the currently monitored instance.
- Specified time range — indicates the period for which data is being displayed (by default: last 60 minutes).
- Instance pane — contains the following indicators:
  - DB Type — identifies whether the monitored database's type is SQL Server, Sybase, DB2, or Oracle.
  - DB Version — identifies the SQL Server version number, along with the most recently installed service packs.
  - Up Since — identifies the date and time when the instance was last started.

NOTE: The pane's initial view only shows the date; to display the time as well, hover the mouse on the instance status icon ( ).

  - OS Version — identifies the operating system's version number, along with the most recently installed builds and service packs.

NOTE: The pane's initial view only shows the OS name; to display the build and service packs as well, hover the mouse or click the OS name's text.

- Response Time — the time (in milliseconds) that elapses from the moment a query, which is supposed to represent the general workload, is submitted, until the application indicates that the query was run.

  Because the response time is usually the starting point for investigation, the Response parameter leads to the SQL Instance Summary panel in the SQL Activity drilldown.

- CPU (%) and Memory (%) — the average CPU load and memory consumption (percentage), during the specified time range, of all CPU units that host the SQL Server instance. This indicator displays the share of SQL Server-incurred CPU load and memory consumption within the total figure.

  Clicking the number on both icons displays a pop-up that shows the total CPU usage or memory consumption on the currently monitored host, plotted over time.

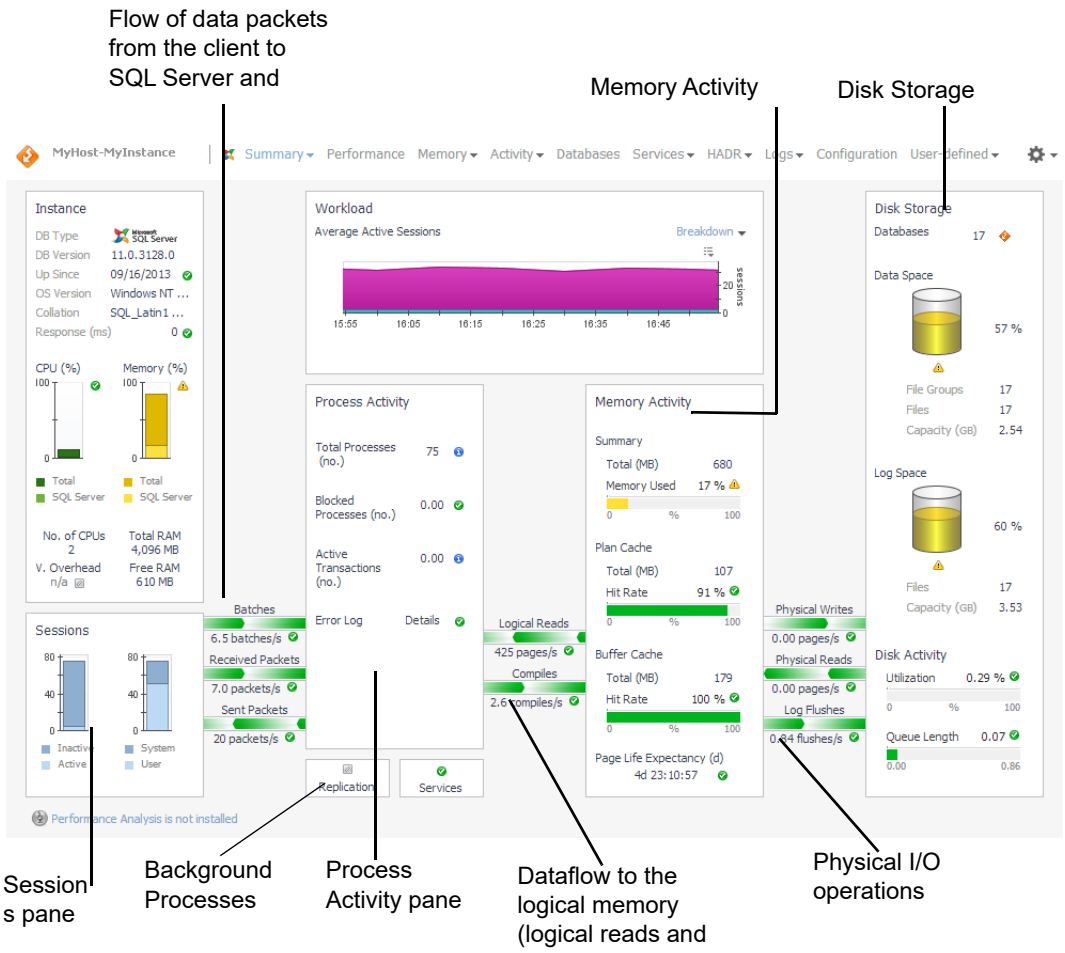
- No. of CPUs — the number of CPU units on the instance's host.
- Total RAM — the total amount (in megabytes) of the host's physical memory.
- V. Overhead — the percentage of CPU that is unavailable to this virtual machine because it is utilized either by other virtual machines or by VMware itself.

  NOTE: The virtualization overhead metric holds value only for virtual hosts running on VMware ESX servers.

  Clicking the virtualization overhead indicator displays the Virtualization Resource Utilization pop-up.

- Free RAM — the total amount (in megabytes) of physical memory available to the applications.

Tracking the Instance Data Flow



The following components represent the data flow in the SQL Server operation, from the session to the physical disk storage:

- The Sessions pane allows examining all sessions running on the instance, distributed between the following session types:
  - Active vs. inactive sessions
  - System (background) vs. User (foreground) sessions

  This pane displays the activity on the client side, including the flow of data (packets and batches of SQL Statements) from the client to the Server and (in the case of packets) back. For details, see Monitoring General Session Statistics on page 77.
- The Process Activity pane — displays several indicators about the instance's sessions, such as the total number of processes and the number of blocked processes. In addition, this pane allows you to track the rate of the data flow used by SQL connections — compiles and logical page reads, that is, referencing pages in the buffer cache. For details, see Data flows.Monitoring total activity on page 77.
- The Background Processes pane — used for tracking the Replication and Services background processes. For details, see Monitoring Background Processes on page 81.
- Memory Activity pane — displays the size of the total SQL memory (both physical RAM and buffer pool), and allows ensuring that the SQL Server appropriately uses its dynamic memory management. The SQL Memory pane's indicators may point out inefficient memory management, which leads to unnecessary use of physical I/O operations. For details, see Monitoring the SQL memory management on page 79.

- Physical I/O operations — displays representations of physical I/O activity. While such activity is necessary to carry out certain operations (for example, accessing a new table), a high level of physical I/O activity may indicate poorly coded SQL or inadequate indexes, which prevent logical reads. For details, see Tracking physical I/O activity on page 80.
- Disk Storage pane — displays the state of the physical disk storage, that is, databases and their backup, data files, and log files. This pane features indicators that refer either to lack of disk space (when a data file is close to reaching its predefined storage ceiling), or performance issues such as high disk queue length.

  In addition, the Disk Storage pane displays the Log Flush wait time alarm, which may indicate an issue with the logical disk activity. For details, see Monitoring the disk storage on page 80.

Monitoring General Session Statistics

The Sessions pane monitors all session types, that is, system, user and SQL Server Agent sessions. Using this pane allows viewing the response time compared with the number of sessions and the instance's level of activity. A high response time value may result from a long queue, that is, an overly high percentage of active users.

Long queues can indicate one of the following issues:

- A massive workload — the system handles more users than it was initially designed to do.
- A bottleneck — lack of system resources prevents users from carrying out their transactions, resulting in wait events and an increasing number of sessions that remain active for prolonged periods.

The Sessions pane displays the total number of sessions, distributed according to the following distinctions:

- Active vs. inactive sessions
- System (background) vs. User (foreground) sessions

The parameters in this pane lead to the **SQL Activity > Sessions** panel. For details, see Reviewing the SQL Server Instance Activity on page 69 and Reviewing Session Details on page 86, respectively, in Reviewing the Instance Activity on page 69.

The client applications represented graphically in the Sessions pane communicate with the SQL Server by sending and receiving network packets and by submitting SQL statements for execution by SQL Server. The flows, detailed in the following table, help indicate performance issues if their values are too low.

Table 15. Data flows.Monitoring total activity

Flow	Description
Batches	The rate at which batches of SQL statements are being submitted to SQL Server for execution.
Received packets	The rate at which the SQL Server receives network packets from client applications. When this icon is yellow, clicking it displays text that describes theBlocking Alarm on page 166 deviations that triggered this display, along with a representation of the rate at which SQL Server is encountering network packet errors.
Sent packets	The rate at which network packets are being sent from SQL Server to client applications. When this icon's color is yellow, clicking it displays text that describes the deviations that triggered this display, along with a graphic representation of the rate at which SQL Server is encountering network packet errors.

The Process Activity pane allows monitoring the processes, both system and user sessions, run by the SQL Server. This pane also features other performance indicators, such as total number of lock requests per second and CPU usage, and allows accessing the error log.

Table 16. The Process Activity pane displays the following parameters:

Parameter	Description
Total Processes	The total number of SQL Server processes, including both user and system processes.
Blocked Processes	Number of processes that are waiting for another process to release a resource that the process is currently locking. Blocked processes can sometimes lead to bottlenecks. The Blocked Process indicator changes its color when one or more processes become blocked. For details, see Blocking Alarm on page 166 and Deadlocks Alarm on page 166.
Active transactions (No.)	The average number of active transactions for each database of the monitored instance.
Error Log	The SQL Server and SQL Agent error logs. When scanning is enabled, Foglight for SQL Server scans the SQL Server logs and raises alarms upon finding error messages that contain any of the error log alert rules. These rules can be specified for all connections, or for the current connection, using the Log Scanning view in the Databases Administration dashboard. For details, see Defining Error Log Filtering on page 156. Hovering over this icon displays the number of errors that were recorded in the SQL Server error log during the selected time range (by default, last 60 minutes). For details about the error log alarms, see Error Log Alarm on page 168.

For further details, see Reviewing the SQL Server Instance Activity on page 69.

To communicate with the SQL memory, the SQL processes use logical reads and compiles, graphically represented as flows in the homepage.

Tracking logical reads

The flow from the SQL Memory pane to the SQL Processes pane indicates the rate at which pages in the buffer cache are being referenced by SQL connections (logical page reads).

Normally, the majority of logical reads is satisfied from the cache, but if the required page is not yet in the cache, it is read from disk.

For details, see Viewing the SQL Server I/O activity on page 71.

Tracking compiles

The Compiles flow displays the rate of SQL compilations and recompilations per second.

Preferably, the most frequently used executable query plans should be retrieved from the procedure cache, thereby saving the need to compile, significantly reducing the utilization of CPU resources, and speeding up the response time of SQL Server queries. Recompiling, which is a CPU-intensive process that may degrade performance, can in certain cases be avoided through sound coding practices.

When many recompiles take place, the SQL Server's CPU can become overloaded, thereby slowing down everything running on that computer. As a result, the Recompiles alarm is invoked. For details, see Recompiles Alarm on page 167.

The Call Rates chart, displayed in the SQL Instance Summary panel of the SQL Activity drilldown, details the use of compiles and re-compiles during the last 60 minutes. For details, see Monitoring the call rates on page 73.

Monitoring the SQL memory management

The Memory Activity pane allows monitoring the SQL Server's dynamic memory management and ensuring its proper handling of the buffer pool.

Each page of memory used by SQL Server is assigned to one of several cache types. Each cache grows and shrinks in size as required. The main caches are the Buffer Cache, which stores a copy of the SQL Server's most recently used database pages, and the Procedure Cache, which holds recently compiled query execution plans. Both these cache types should satisfy I/O requests and save the need for physical reads from the disk. The Buffer cache's efficiency also affects the Page Life Expectancy, another indicator displayed in the SQL Memory pane.

Table 17. The SQL Memory pane displays the following parameters:

Summary section	
Total (MB)	The total amount of memory, either fixed or dynamically allocated, which the SQL Server is currently using. When the amount of SQL Server memory available for immediate reuse drops below a threshold, the Free Buffers alarm is invoked. For details, see Free Buffers Alarm on page 169.
Memory Used	Displays the total amount of memory, either fixed or dynamically allocated, that the SQL server is currently consuming, scaled against the maximum amount of memory it can use. By default, SQL Server manages its total memory automatically, adjusting it to the varying memory requests of both SQL Server processes and Windows processes. Clicking this gauge allows accessing the Memory > Buffer Cache panel, which provides an in-detail view of SQL Server's memory management. For details, see Viewing the Memory Summary on page 62.
Plan Cache Section	
Total	Displays the amount of memory currently allocated to the plan cache (formerly known as procedure cache), a memory area used by SQL Server to hold recently compiled query execution plans stored procedures, triggers, ad hoc SQL, and so on. The value of this metric is calculated as follows: <pages> * 8K / 1024
Hit Rate	Displays the percentage of plan cache lookups that found the required plan already in the cache. If a matching plan is found, SQL Server does not need to compile the query/stored procedure. This can save a significant amount of CPU resources, and can speed up SQL Server queries. A low Plan Cache hit rate may lead to performing extra compilations, thereby degrading SQL Server performance by causing extra CPU load. In a such situation, the Procedure Cache Hit Ratio Alarm is invoked. For details, see Procedure Cache Hit Ratio Alarm on page 170. The plan cache hit rate is the average hit rate for all procedure cache object types, except for ad hoc SQL. This rate is calculated using a differential sampling method, which gauges only the last few sampled periods.
Buffer Cache Section	
Total	Displays the amount of memory currently allocated to the buffer cache, including database, free, and stolen pages. The buffer cache, which is an in-memory copy of recently used database pages, is normally the largest memory cache used by SQL Server. If an SQL process needs to access a database page, finding this page in the buffer cache spares the SQL Server the need to read the page from disk, thereby significantly reducing the amount of disk I/O and speeding up queries. The value of this metric is calculated as follows: <pages> * 8K / 1024

Table 17. The SQL Memory pane displays the following parameters:

Summary section	
Hit Rate	Displays the percentage of database page I/O requests that were satisfied from the buffer cache. High buffer cache hit rate indicates that SQL Server efficiently uses the memory allocated to its buffer cache. Low buffer cache hit rate, on the other hand, indicates that SQL Server is finding fewer pages already in memory, resulting in the need to perform more disk reads. Low buffer cache hit rate invokes the Buffer Cache Hit Ratio alarm. For details, see Buffer Cache Hit Ratio Alarm on page 168.
Page Life Expectancy	Displays the current Page Life Expectancy, that is, the length of time in seconds that a database page stays in the buffer cache before it is flushed out. Small values indicate that the buffer cache retains pages for short periods, as a result of high data cycling, and that the buffer cache is not being effective. Microsoft recommends 300 seconds as the minimum value for this metric; any less is indicative of a shortage of memory. For details, see Page Life Expectancy Alarm on page 169.

Tracking physical I/O activity

The physical I/O Operations section of the Foglight for SQL Server Instance Homepage displays graphical representations of the number of pages physically read from and written to disk by SQL Server.

In addition, this section allows viewing the SQL Server processes Checkpoint and Lazy Writes.

Parameter	Description
Physical Writes	A flow that represents data/index pages written to disk per second. Normally, SQL users do not have to wait for database write operations to complete, as most modifications to database pages are made in the buffer cache.
Physical Reads	A flow that represents page reads from the disk per second. Physical reads are used when a connection requests a page that is not already in the buffer cache. Physical read operations are necessary when accessing new table or index pages. Nevertheless, these operations should be avoided when possible, as requested pages should reside in the SQL Server buffer cache.
Log Flushes	A flow that represents the number of log pages per second being written to disk by the Log Writer process. For details, see the glossary definition of Log writer on page 296.

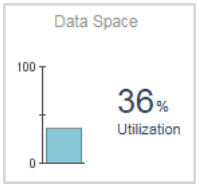
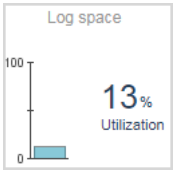
For details, see Monitoring the SQL Server physical I/O activity on page 82.

Monitoring the disk storage

The Disk Storage pane displays the status of the storage devices in the SQL Server system, that is, databases and their backup, data files, and log files. Using this pane allows identifying storage issues such as a database that has not been backed up or a data file that has almost exhausted its growth potential and is going to fill completely.

In addition, the Disk Storage pane displays the Log Flush wait time alarm, which may indicate excessive I/O operation.

Table 18. The Disk Storage pane.

Parameter	Description
Databases	The total number of available databases in the SQL Server instance. This icon displays an alarm when one or more of the databases becomes unavailable, or if any database has not been backed up in the last few days. For details about handling these alarms, see Recent Backups Alarm on page 173 and Database Unavailable Alarm on page 173.
Data Space	The total number of data files in all databases of the SQL Server instance. The disk icon represents the total percentage of currently used space within the entire capacity of all data files that reside on all databases. The total percentage is also displayed in text at the bottom of this section, below the text that displays the physical size of data files space used. 
Log Space	The log files in all databases of the SQL Server instance.The total percentage is also displayed in text. 
Disk Activity section	
Disk Utilization	The percentage of time the busiest disk spent serving system-wide I/O requests. The Disk Utilization metric serves as a measure for the system I/O load. High values may indicate a device bottleneck, due to either disk fragmentation or I/O resource contention of multiple processes that try to write or read from the disk.
Disk Queue Length	The average number of I/O requests that are queued and waiting for an available disk during the sample interval. This figure may include I/O activity generated by both SQL Server and non-SQL Server processes. For details, see Viewing the disk queue length on page 83.

Monitoring Background Processes

The Background Processes pane displays the status of optional SQL Server components, such as the Replication Agent, which is implemented as a job of SQL Server Agent service, and several of the SQL Server services.

Some of the supporting services significantly enhance the database efficiency and accuracy, and therefore should be active at all times; for example, DTC, which ensures successful and complete transactions.

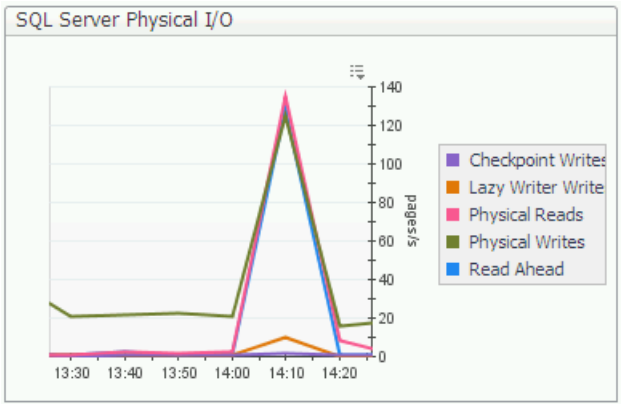
Table 19. The Background Processes pane.

Indicator	Description
Replication	Indicates whether SQL Server Replication is configured. If replication is configured, indicates whether there are replication errors. Replication is an optional component of SQL Server, which synchronizes data between different SQL Server databases.
Services	Status of the SQL Server services that are currently monitored. These services are supplementary to SQL Server and provide capabilities such as mail and full text search. This icon invokes an alarm when any of the SQL Server services are installed, but not active. For details, see Alarms Displayed in the Background Processes Panel on page 170. For details about the drilldowns used for handling the supporting services alarms, see Reviewing the Support Service Status on page 113.

Monitoring the SQL Server physical I/O activity

The *SQL Server Physical I/O* chart displays the rate at which pages are physically read from and written to disk by SQL Server. The read and write operations are carried out by either SQL Server processes or system processes.

Figure 20. The SQL Server Physical I/O chart.



Physical read operations are necessary when creating a table or an index page for the first time. Nevertheless, these operations should be avoided when possible, as requested pages should reside in the SQL Server buffer cache. High rates of these metrics indicate extensive data operations.

The following list presents the main physical I/O types shown in the chart:

- *Checkpoint Writes* are the most common type of write activity under normal circumstances. The checkpoint process periodically scans the buffer cache for modified pages and flushes all modified pages out to disk, thereby minimizing the amount of work SQL Server is required to do on restart.

  Checkpoint writes are carried out at intervals, which are defined by the recovery interval parameter. If this parameter's value is too high, the checkpoint process may run infrequently, thereby overloading the Lazy Writer process. As a result, the Lazy Writer does not efficiently maintain the Free Pages list and, when a certain threshold has been exceeded, the Free Buffers alarm is invoked. For details, see Free Buffers Alarm on page 169.

- *Lazy Writer Writes* are carried out when the Lazy Writer process needs to free up buffer pages that have been modified in the buffer cache. Freeing up the buffers requires the Lazy Writer process to write first the changed pages to disk.

  High value of lazy writes may indicate that SQL Server is running out of available space in the buffer pool cache. Use this chart to view whether the high lazy writes value is a consistent issue.

When the amount of SQL Server memory available for immediate reuse drops below a certain threshold, the Free Buffers alarm is invoked. For details, see Free Buffers Alarm on page 169.

- *Physical Page Reads* are carried out when a user connection requests a page that is not already in the buffer cache. The connection requesting the page awaits until the I/O operation completes.
- *Physical Page Writes* are write operations where the user connection has to wait for the I/O to complete before continuing. These are most often caused by operations such as create index, bulk insert, or restore.
- *Read Ahead* occurs when SQL Server forecasts the need for data that currently resides on the disk. In this case, the pages are pre-fetched into the buffer cache before being requested by the user, using Read Ahead processing.

Monitoring the SQL Server logical I/O activity

The *SQL Server Logical I/O* chart displays the number of Logical reads (getpage requests) issued by SQL Server.

SQL Server keeps a copy of its most recently used database pages in the buffer cache. When a connection needs to reference a database page, SQL Server performs a Logical I/O operation by checking the buffer cache to see if the requested page is already in memory. If the page is found in the buffer cache, a Logical I/O read is carried out; otherwise, the page is read from disk, using a Physical I/O operation.

Viewing the disk queue length

The *Disk Queue Length* chart displays the disk activity for each logical disk.

Disk Queue length tracks the average number of I/O requests that are queued and waiting for an available disk during the sample interval. This figure may include I/O activity generated by processes other than SQL Server. Values that exceed the threshold set in this metric may indicate a system bottleneck.

NOTE: Storage Area Network (SAN) storage array may indicate high Disk Queues for sustained periods, even though no disk I/O bottleneck takes place. Therefore, it is advisable to check related information about read, write, and transfer operations, and determine whether there are pending SQL Server I/O operations, before assuming the occurrence of I/O bottlenecks.

Viewing how SQL Server's logical data is accessed and updated

The *Access Methods* chart allows viewing various statistics used for monitoring the methods used for accessing and updating SQL Server data. These counters are all collected from the SQL Server Access Methods Manager.

Figure 21. The Access Methods chart

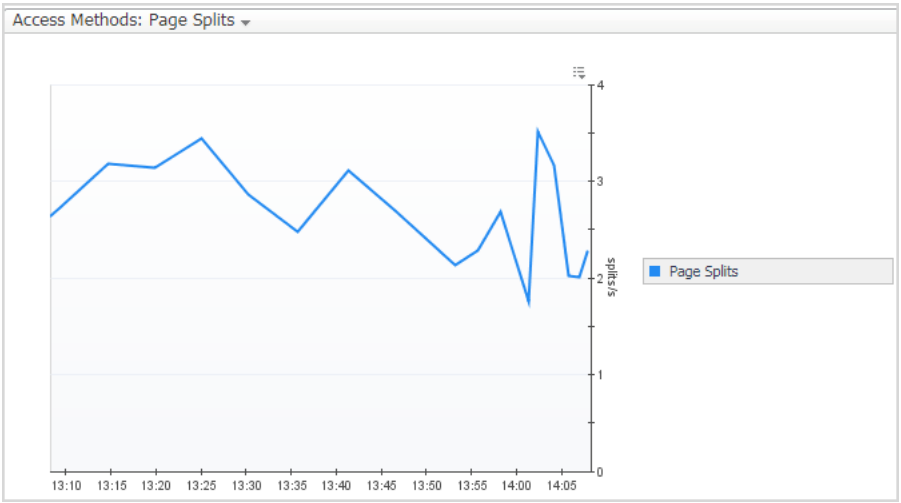


Table 20. The Access Methods list allows you to choose one of the following counters:

Counter	Description
Page Splits	Displays the rate at which index database pages are being split into two because the page does not contain enough free space to accommodate an inserted or updated record.
Page Allocations	Displays the rate at which pages are being allocated to tables or indexes. Indicates how fast tables are expanding.
Page Deallocations	The rate at which pages are being de-allocated from tables or indexes, thereby indicating how fast tables are shrinking.
Freespace Scans	The number of scans initiated to search for free space to insert a new record. A high rate of freespace scans can indicate disk fragmentation.
Forwarded Records	The rate at which SQL Server is retrieving forwarded records. When a row in a table that *does not* have a clustering index is updated - and if the modified row no longer fits on the old page - SQL Server moves the row to a new page and leaves a forwarding pointer in the old page. Using this method saves the need to modify the non-clustered indexes in order to reflect the new row location; however, when this method is implemented, subsequent retrieval of this row using non-clustered indexes requires extra I/O resources. A high Forwarded Records rate can indicate a need to reorganize the existing tables (unload/reload) or define clustered indexes.
Ghosted Records	The rate at which SQL Server encounters ghosted records during scans. When a record is deleted from a table, SQL Server improves concurrency by *not* physically removing the row from index Leaf levels, but marking it instead as deleted (ghosted). At some later point, a housekeeping process asynchronously removes these rows from the leaf level. Until the records are removed, SQL Server must skip the ghosted records during leaf-level scans.
Table Lock Escalation	The rate at which locks are being escalated to the table level. A high table lock escalations rate may indicate a need to tune queries, in order to avoid designing multiple page locks that continually force SQL Server to carry out such escalations.

Viewing Session Details

The Sessions panel allows you to trace the activity of all currently opened sessions, as well as their resource consumption.

The Sessions panel is divided into 3 views:

- Resource consumption - displays the current and last 1 hour resource consumption.
- Current - displays all the current opened sessions.
- Last 1 Hr. - Enabled only when SQL PI is configured. This view provides several significant features, which highly enhance data retrieval and display capabilities, viewing the top sessions that were executed in the instance during the last 1 hour and filtering them based on the resource consumption.

Reviewing Resource Consumption

The resource consumption section is divided into 2 time frames, Current and Last Hour.

The Current section allows viewing the consumption of various resources of all the sessions currently opened to the instance. The resource list is as follows:

- **Sessions** — Displays the distribution of the currently opened SQL Server sessions between active (sessions that are actively processing in SQL Server) and inactive sessions during the current sampling period.
- **Throughput** — Divided to 3 metrics which represent the throughput of the monitored instance:
  - Sessions/Sec — The rate per second of sessions that were initiated during the current sampling period.
  - Transaction — The average number of active transactions on the monitored instance during the current sampling period.
  - Batches/Sec — The rate per second of statement executions during the current sampling period.
- **Blocked** — Displays the total number of SQL Server sessions that are blocked during the current sampling period.
- **CPU** — The Host CPU Usage the total percentage of CPU resources being used on the monitored host. The displayed value represents the total CPU that is consumed by all Windows processes (SQL Server and non-SQL Server processes).

The Last 1 hour trend presents the activity on the instance during the last 1 Hour by presenting:

- Workload — displays the general workload (average active sessions) during the specified time range.
- Breakdown — displays the workload, distributed by the various wait event categories. The colors of the categories match the colors of the various resources displayed in the Resource Breakdown section.

By clicking the **All Wait Events** link at the upper right corner of the section, The *Active Wait Events* pop-up appears, with the Resource drop-down list unfiltered and displaying the entire list of wait events

Reviewing Current session details

The current sessions view allows you to monitor the currently running sessions and their individual resource consumption. The Session List section contains a table that lists all currently running SQL Server sessions, displaying each session in a separate row. The Sessions view in the table can be filtered by the Active only and Foreground Only check boxes:

- Active only — selected by default, presenting only the active sessions that are currently running, by clearing the check box, all inactive sessions will be presented as well.
- Foreground only — selected by default, presenting only the sessions whose SPID number is higher than 50, by clearing the check box, the background sessions will be presented as well which holds the instance sessions ID's 1 through 50.

Table 21. The current Sessions table contains the information detailed below.

Column	Description
Kill	Allows termination of the selected session, using the *Kill Session* dialog box that is displayed upon clicking this column.
SPID	The session unique identifier (Session Process ID).
Status	Indicates whether the session is active, blocked or suspended.
DB User	The SQL Server login name for this session.
Database	The name of the database the user is accessing.
CPU Usage	Time spent by the various sessions consuming CPU cycles. The CPU usage value is read directly from the operating system.
Memory Usage	The portion, in megabytes, of the procedure cache allocated to this process. A negative number indicates that the process is freeing memory allocated by another process.
Active Time	Sum of all the active wait events, equal to the session total activity within the current interval.
Transaction Count	The number of open transactions. This parameter's value corresponds to the value of the session's `@@trancount` — a global variable that reflects the level of nested transactions.

Table 21. The current Sessions table contains the information detailed below.

Column	Description
Last batch	The latest batch ran by this process.
Login Time	The time when the session was created.
Program	The program that is the owner of this SQL Server session.
Host Name	The name of the client computer that established the SQL Server connection.
Last Batch Time	The time when the last batch started execution.
Blocked by	Which SPID (if any) holds locks on a resource on which this session is waiting.
Cache Hit Ratio	The percentage, for this session, of buffer cache hit ratio, that is, file read operations that were satisfied by the file system cache without requiring any physical I/O. The value of this metric should be as high as possible.
Physical Writes	The total number of data/index pages written to disk by the selected session Normally SQL users do not have to wait for database write operations to complete. Most modifications to database pages are made in the buffer cache.
Logical Reads	The total number of logical reads carried out by the selected session.
Physical Reads	The total number of physical reads carried out by the selected session.
Activity	Indicates whether the session is active.
Context Info	The context information of the current session.
Last Command	The currently executed or most recently executed command.
Lock Timeout	The amount of time that this session will wait for lock requests to be satisfied; corresponds to the @@Lock_Timeout global variable for this session.
Session Up Time	The amount of time since the session is started (dd:hh:mm:ss).
Resource Pool name	The name of the resource pool attached to the session
XTP Transactions	The number of In-Memory OLTP (XTP) open transactions by the session
Oldest Transaction Duration	The duration (in seconds) of the oldest transaction opened by the session

NOTE: To accelerate the sessions' retrieval time, only the first 60 characters of the SQL text of each query are being retrieved.

The refresh rate of the data displayed on the Sessions panel can be set by selecting a value from the Refresh interval drop-down list, which appears on the panel's upper right side.

Reviewing Session Details

The various sections of the Session Details page provide detailed information about the selected session upon selecting a specific session in the table.

Configuring SQL PI presents in-depth analysis of the session details. See Reviewing Session details with SQL PI on page 89

Session Identification Section

Table 22. Displays several parameters that provide general information about the selected session, as follows:

Parameter	Description
Session ID	Session Process ID; the unique number that SQL Server has assigned to identify the selected session. **NOTE:** Clicking on the Session ID enables you to switch sessions from a pop-up list of current sessions.
Login Time	The time when the user logged on to the session.
Status	Indicates whether the session is active or inactive.

Table 22. Displays several parameters that provide general information about the selected session, as follows:

Parameter	Description
DB User	The SQL Server login name for this session.
Program	The name of the program the user is running to access SQL Server.
Machine	The name of the host the session is running from.
Waiting for:	The resource on which the session is waiting.

Reviewing Workload details

The Workload page displays the selected session workload (Seconds/Sec) during the specified time range (default 1 Hr.). The breakdown displays the workload, distributed by the various wait event categories. The colors of the categories match the colors of the various resources displayed in the Resource Breakdown section. By clicking the **All Wait Events** link at the upper right corner of the section, The *Active Wait Events* pop-up appears, with the Resource drop-down list unfiltered and displaying the entire list of wait events.

CPU Section

Displays the sum amount f CPU (in seconds) that the selected session is consuming.

IO Section

Displays the sum amount of IO (in pages) that the selected session is consuming.

Session Details Section

The Session Details section contains several tabs of information.

Reviewing the Most Recent SQL details

This tab provides details about the statement that is currently being executed by the selected session.

Table 23. This pane contains a table with the following columns:

Column	Description
SQL Batch	The full text of the SQL execution batch.
CPU Time	Total CPU time consumed by the statement
Physical IO Reads	The total number of physical I/O reads performed by the selected SQL statement.
Physical IO Writes	The total number of physical I/O writes performed by the selected SQL statement.
Active Time	Summary of all Sum of active wait events incurred by the SQL statement.
Logical Reads	The total number of logical reads carried out by the selected SQL statement.
Waiting On	The resource on which the session is waiting.

Reviewing Metrics

Additional columns are displayed in the Metrics table when SQL PI is configured. See Reviewing Metrics details with SQL PI on page 90.

Table 24. The Metrics pane contains the sum of activity for the selected session in a table that displays following metrics:

Column	Description
CPU Usage	Time spent using CPU. Calculated in seconds.
Transaction Count	Number of open transactions for the session.
Physical Reads	Total number of physical reads operations performed by all requests running in the instance.
Physical Writes	Total number of logical write operations performed by all requests running on the instance.
Logical Reads	Total number of logical read operations performed by all requests running on the instance.
Cache Hit Ratio	The ratio of logical reads to physical reads. It indicates the percentage of database page I/O requests that were satisfied from the Buffer Cache and therefore did not have to perform disk reads
Memory Usage	Amount of memory
Active Time	Sum of all the active waits and CPU usage, equal to the session total activity within the current interval.

Reviewing Session Trace details

The Session Trace pane uses SQL Server trace functionality to display the SQL events activity generated by the selected session. The retrieval of the data displayed in the Session Trace pane starts upon entering the pane, and retrieval continues throughout the session investigation run by Foglight for SQL Server. Two minutes after exiting the session pane, data retrieval for the specific session is automatically disabled.

Table 25. The Session Trace pane allows you to trace the following events:

Event	Description
RPC :Completed	Occurs when a remote procedure call (RPC) has completed
SQL :Batch Completed	Occurs when a Transact-SQL batch has completed.
Event Class	The type of class that was traced.
Start Time	The time when the class event started.
End Time	The time when the class event ended.
Duration	The duration of the class event in milliseconds.
Text Data	The query's text. This column displays data only if the event class or classes being captured by the trace contain text.
Reads	The number of logical disk reads performed by the server on behalf of the event.
Writes	The number of physical disk writes performed by the server on behalf of the event.
CPU Usage	The amount of CPU time (in milliseconds) used by the event.
Database ID	ID of the database currently used by the session.
Object Name	System-assigned ID of the object.
Index ID	ID for the index on the object affected by the event.

Reviewing Session Blocks details

The Session Blocks pane displays all blocks held or requested by the selected session. The Lock Status column indicates whether the locks have been granted to the session, or are currently blocked and waiting on another session.

Table 26. The Session Blocks table displays the following columns:

Column	Description
Session ID	The session unique identifier (Session Process ID). The unique number that SQL Server has assigned to identify the selected session.
SQL User	The SQL Server login name for this session.
Type	The type of the currently locked resource (Database, table, page, row, extent, and more).
Status	The status of the lock, for example, running or blocked.
Wait time	The time the session is waiting on lock.
Resource	The name of the currently locked object.
Command	The currently executed command.
Program	The name of the program the user is running to access SQL Server.
Win user	The Windows login name for this session.
CPU	The amount of CPU time, in milliseconds, used by the event.
IO	The total number of I/O operations performed by the selected SQL statement.
Host Name	The name of the host the session is running from.
DB Name	The database on which the lock is taking place.
Session SQL	The full text of the SQL execution batch.

Reviewing Session Locks details

The Locks table displays information about all locks currently held or requested.

Table 27. The Locks table displays the following columns:

Column	Description
SPID	the server process ID of the current user process.
Database	The database being used by the current process.
Count	The number of locks of the type specified in the Lock Type column against the database specified in the Database column, from the SPID specified in the SPID column.
Index Name	The index being used by the current process, if any.
Object Name	The currently locked object.
Lock Type	The type of the resource that is currently locked. For example, database, table page, row, or extent.
Mode	The kind of lock being applied to the resource. For example, shared, exclusive, update, IntentShared, or IntentExclusive.
Status	The status of the lock: Blocked, Blocking, or blank (Granted)

Reviewing Session details with SQL PI

Configuring SQL PI enhances the Current session details by adding the next enhancements:

Reviewing SQL Summary details

The SQL Summary pane displays summarized data about each of the SQL Statements that were executed by the session. The SQL statements are sorted into a table that represents the activity of each SQL statement that ran or still running on the selected session.

Table 28. The SQL Summary table displays the following columns:

Column	Description
SQL Statement	The SQL statement's text.
CPU Usage	Total CPU usage consumed by the statement.
CPU Wait	Time spent by waiting in the systems run queue for CPU cycles. This reading is calculated from the operating system readings, rather than the SQL Server wait states.
I/O Wait	Time spent waiting for disk input/output operations to complete.
Memory Wait	Time spent waiting by the various processes waiting for the completion of a log operation.
Network Wait	Network wait events occur when a session spends time waiting for messages to be sent or received over the network interface.
Lock Wait	Time spent waiting for a blocking lock, held by another session, to be released.
Log Wait	Time spent waiting for a Log operation to complete.
Current Wait Time	The number of seconds SQL Server has spent waiting for I/O operations on this file since the last time data was collected.
Latch Wait	Time spent waiting for internal locks (latches) to be released.
CLR Wait	Time spent waiting for CLR code execution to complete.
Remote Provide Wait	The time waiting for a remote OLEDB call to complete or DTS synchronization.
Other Wait	Time waiting for miscellaneous log waits.
SQL Batch	The full text of the SQL execution batch.
Elapsed Time	The amount of time, in seconds, that the SQL statement's execution lasted.
Physical IO Reads	The total number of physical I/O reads performed by the selected SQL statement
Physical IO Writes	The total number of physical I/O writes performed by the selected SQL statement.
Logical Reads	The total number of logical reads carried out by the selected SQL statement.
Command Type	The type of the command that was carried out by the selected SQL statement. For example, insert, select.

The Activity Panel displays the SQL Batch that is executed for the selected SQL statement in the table and a pie chart which display the SQL statement workload activity.

Reviewing Metrics details with SQL PI

Selecting a metric row from the metric list present this metric activity in a chart. The data displayed in the chart represents the metric activity over the selected time frame.

Table 29. The following columns are added the metrics table when SQL PI is configured:

Column	Description
Backup Recovery	Time spent waiting for backup or recovery tasks to complete.
Cursor Synchronization	The time spent synchronizing information flow within cursors.
Database Replication	Time spent waiting for replication synchronization events to complete.
Deferred Task Worker	Time spent waiting for I/O requests.

Table 29. The following columns are added the metrics table when SQL PI is configured:

Column	Description
Distributed Transaction	Time spent waiting for various distributed transaction events to complete.
External Procedure	Time spent waiting for external procedures to end.
Full Text Search	Average CPU consumption by the Full Text Search service.
Hosted Component	Time spent waiting for hosted components such as, but not exclusively, CLR.
I/O Bulk Load	Time spent waiting for the completion of I/O operations required to carry out a bulk load I/O.
I/O Completion	Time spent waiting for I/O operations to complete.
I/O Data Page	Time spent waiting to latch a buffer for an I/O request.
Internal Cache Latch	Time spent waiting for latches that are not buffer latches or savepoint latches.
Latch Buffer	Time spent waiting to latch a buffer that is not an I/O request.
Latch Savepoint	Time spent waiting to synchronize commits to marked transactions.
Lock Bulk Update	Time spent waiting to acquire bulk update locks.
Lock Exclusive	Time spent waiting to acquire exclusive locks.
Lock Intent	Time spent to acquire intent locks.
Lock Schema	Time spent waiting to acquire schema locks
Lock Shared	Time spent waiting to acquire shared locks.
Lock Update	Time spent waiting to acquire update locks.
Log Buffer	Time spent waiting for space in the log buffer or otherwise waiting for memory to be made available to write log records.
Log Other	Time spent waiting for miscellaneous log.
Log Synchronization	Time spent waiting to see whether log truncation frees log space.
Log Write	Time spent waiting for outstanding I/O to finish, or waiting for log flushes to complete.
Network HTTP	Time spent waiting for outstanding HTTP connections to complete and exit.
Network I/O	Time spent waiting for network packets.
Network IPC	Time spent waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.
Network Mirror	Time spent waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.
OLEDB Provider Full Text	Time spent waiting for the Microsoft Full Text Engine for SQL Server.
Other Miscellaneous	Time spent waiting for miscellaneous database operations.
Parallel Coordination	Time spent waiting for parallel coordination tasks to complete.
Service Broker	Time spent waiting for Service Broker event handlers and endpoints.
Synchronous Task	Time spent waiting for tasks started synchronously. Most SQL Server processes are started Asynchronously.

Reviewing Session details

Displays the same data as described in the Reviewing Session details with SQL PI on page 89.

The Sessions details can be sorted by selecting a resource from the drop down resource list above the left panel, by selecting the resource the workload chart will be updated accordingly.

NOTE: The Session Blocks and Session Locks tabs are not presented when drilling to the session's details from the Last 1 Hour view.

Monitoring Locks and Latches

The **Locks** panel displays information about all locks and latches in the currently monitored SQL Server instance.The information is provided by the panes described in the following topics:

- Locks table
- Lock Types chart on page 92
- Latches chart on page 93

Locks table

The *Locks* table displays information about all locks currently held or requested.

To configure the default retrieval settings for this panel, use the Lock view in the Databases Administration dashboard. For details, see Setting Options for Displaying Data in the Locks Panel on page 159.

Table 30. The *Locks* table displays the following columns:

Column	Description
SPID	The server process ID of the current user process.
Database	The database being used by the current process.
Count	The number of locks of the type specified in the Lock Type column against the database specified in the Database column, from the SPID specified in the SPID column.
Index Name	The index being used by the current process (if any).
Object Name	The currently locked object.
Lock Type	The type of the resource that is currently locked (Database, Table, Page, Row, Extent, and so on).
Mode	The kind of lock being applied to the resource (Shared, Exclusive, Update, IntentShared, IntenExclusive, and so on).
Status	The status of the lock: Blocked, Blocking, or blank (Granted).
Login Name	SQL Server login name for this session.

Lock Types chart

The *Lock Types* chart shows lock statistics broken down by the various types of locks available in SQL Server.

This chart displays one line for each of the following lock types:

- AllocUnit — a lock on an allocation unit
- Application — a lock on an application-specified resource
- Database — a lock on a database, including all of the database's objects
- Extent — a lock on a contiguous group of eight pages
- File — a lock on a database file
- HoBT — a lock on a heap of data pages, or on the BTree structure of an index
- Key — a lock on a row in an index
- Metadata — a lock on a piece of catalog information, also called metadata

- Object — lock on table, stored procedure, view, and so on, including all data and indexes. The object can be anything that has an entry in sys.all_objects.
- Page — a lock on an 8-kilobyte (KB) page in a database
- RID (Rows) — Row ID; a lock on a single row in a heap

Use the list on the chart title to select which of the following lock types to display:

- **Waits** — the rate of lock request wait events. Such wait events take place when lock requests cannot be satisfied immediately and require the caller to wait before being granted the lock.
- **Average Wait Time** — the average time (in seconds) that elapses before a lock request wait is cleared.
- **Lock Requests** — the number of lock requests and lock conversion requests per second.
- **Timeouts** — the number of lock time-outs per second.

  By default, SQL Server never times out locks. However, many applications issue a SET LOCK_TIMEOUT statement to cause SQL Server to time out their locks after the specified interval. This metric shows how often these time-outs are being exceeded.

  The Lock Timeout (@@LOCK_TIMEOUT) values for each connection are displayed in the **Session Details** pane, in the **Sessions** panel. For details, see Viewing Session Details on page 84.
- **Deadlocks** — the number of lock requests per second that resulted in a deadlock.

  A deadlock occurs when multiple SQL Server sessions request conflicting locks in such a way that two locks are blocked by each other. For further details, see Deadlocks Alarm on page 166.

Latches chart

The *Latches* chart displays statistics on latch requests.

This chart shows the following series of data:

- **Latch Waits** — how many wait events for latches occurred in the specified time range
- **Total Wait Time** — the total amount of time (in milliseconds) that latch requests spent waiting during the specified time range

Latches are file system locks, used for synchronizing data within SQL Server. Latches are enforced when a data element is being accessed physically, in order to ensure that the data page on which the data element resides is readable and writable.

Enforcement of latches is also carried out before the modified data page is written to disk, to prevent modifications by other users during the physical write operation. After the page is successfully written to disk, the latch is released.

A high rate of latch wait events per second may indicate a slow disk I/O subsystem.

Tracking Current Lock Conflicts

The **Blocking (Current)** panel provides details for all current lock conflicts.

This panel allows carrying out the tasks detailed in the following topics:

- Handling Blocking Sessions, using the *Blocking* table. For details, see Handling blocking sessions.
- Monitoring the processes blocked during the reporting period, using the *Number of Blocked Processes* chart. For details, see Monitoring blocked processes for the sampled interval on page 94.

Handling blocking sessions

The *Blocking* table displays all connections that are either currently waiting on locks held by others, or are causing others to wait, highlighting who is waiting on whom, and the resources involved.

To create a custom filter for this table, use the options accessible by clicking the **Customizer** button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

The hierarchy in this tree diagram represents the blocking chains. It shows who is blocking whom, by displaying one entry for each session that is blocked, and one for each session that is blocking another but is not blocked itself. Sessions at the top of the tree (those that do not have a *parent* in the tree) are at the head of the blocking chain, and are therefore the root cause of all blocking. Such sessions appear as *Lead Blockers* in the *Number of Blocked Processes* chart.

Table 31. The *Blocking* table displays the following parameters:

Parameter	Description
SPID	The unique number the Server has assigned for identifying the selected session.
Wait Time	How long this session has been waiting for the lock (measured in seconds). If the value displayed is 0, the session is not waiting.
Type	The type of the lock request that is waiting (Database, Table, Page, and so on).
Resource	The resource that is in conflict. This value often identifies a database and table. The data in the Resource column is reported directly from SQL Server.
Command	The current or previous command executed. This information can be useful when deciding which sessions to kill.
SQL User	Identifies the user associated with the SPID.
Program	The application program that the user is using (for example, Microsoft Access).
Win User	Name of the Windows account with which the user is logged in to SQL Server.
CPU	The total amount of CPU consumed by the session so far.
I/O	The total amount of I/O resources consumed by the session so far. This information can be useful when deciding which sessions to kill.
Host Name	The name of the client computer.
Status	The status of the session (Blocked, Blocking, or both). For sessions at the head of the blocking chain (those that are not blocked), this will indicates if the session is Runnable or Sleeping.
DB Name	The name of the database where the session is active.
Session SQL	Displays the SQL belonging to the session that is blocked and/or blocking.

Monitoring blocked processes for the sampled interval

The *Number of Blocked Processes* chart displays the number of SQL Server sessions that were involved in blocks over time. Use this chart to review the frequency and duration of lock conflicts in SQL Server.

This chart displays the following indicators:

- **Blocked Processes** — number of sessions that were waiting on locks held by others.
- **Lead Blockers** — number of sessions that were not blocked, but were blocking others. Lead Blockers correspond to sessions in the Blocking table that do not have a parent in the Blocking chain (at level 1 in the tree).

Tracking Deadlocks and their Affected Objects

The Deadlocks panel displays all deadlocks that took place within the selected time range.

Deadlock situations are highly time and resource-consuming, and result in unresponsive applications and operation rollback, because they involve at least two transactions that lock one another and can be resolved only by terminating one of the transactions, making it a “deadlock victim”. Resolving a deadlock can become even more complicated when the deadlock involves more than two sessions (chained deadlocks).

Using the Foglight for SQL Server agent for collecting Deadlock Graph trace data does not require turning on any SQL Server trace flags. Foglight for SQL Server collects this data implicitly for any SQL Server user who has been granted the ALTER TRACE permission through the automatic discovery wizard; the user does not need to intervene in the process.

To retrieve Deadlock Graph data, Foglight for SQL Server uses minimal trace over deadlocks, collecting only the deadlock-related data, unlike SQL Server Profiler. The only difference is in the refresh rate; while SQL Server Profiler constantly refreshes its data, Foglight for SQL Server agent collects the data every 1 minute (by default) when focusing the screen (online mode), and every 5 minutes when off screen (offline mode). As a result, when watching the deadlock dashboard the data refresh occurs every 1 minute (by default).

The Deadlocks panel contains an overview pane, which provides significant performance-related details about each of the deadlocks, such as the amount of time the “deadlock victim” ran before being terminated, the amount of time each of the transactions involved in the deadlock had to wait, and the ratio of regular to chained deadlocks during the specified time range. The panel’s other panes focus on specific types of components (databases/objects/applications), allowing to view which of the application’s components were most adversely affected by deadlocks, and are therefore the most vulnerable to deadlock situations.

This panel includes the panes described in the following sections:

- Deadlocks
- Databases on page 96
- Objects on page 96
- Applications on page 96

Deadlocks

The Deadlocks pane contains the following components, which provide information over all of the deadlocks that took place during the specified time range:

- Summary — provides summarized data regarding the following aspects:
  - Overall lost time — the total amount of time that the terminated session (“deadlock victim”) was running before its termination.
  - Number of deadlocks — the total number of deadlock situations that took place during the specified time range.
- Chained and Regular Deadlocks — contains the following components:
  - Chart — displaying the distribution of locks, with different color codes for chained and regular deadlocks.
  - Graph — displaying the number of regular and chained deadlocks
- Table — displaying detailed data about each of the deadlocks, as listed below.

Table 32. Deadlock details include:

Column	Description
Time	The exact time when the deadlock took place.
Type	The deadlock type: • Chained — indicated by a lock sign • Regular — the column is empty
SPID	The SQL Server’s internal process ID
SQL Text	The short text of the selected SQL statement.
Database	Database name
Host Name	The name of the client computer that established the SQL Server connection.
Login Name	The SQL Server login name for this session.
Client Application	The client application that established the SQL Server connection.
Lost Time	The time, in seconds, since the last transaction ran until the deadlock occurred.
Last trans. started	The time when the last transaction started.
Wait Time	The amount of time the process is waiting for the resource.
Log Used	The amount of log used through the session process.

Table 32. Deadlock details include:

Column	Description
Lock Mode	The lock mode set for the Owner.
Trans. Count	The number of opened transactions.
Owned Object	The object that is owned by the selected session and waited by the other session involved in the deadlock.
Waited Object	The object that is owned by the other session involved in the deadlock, and for which the selected session waits.
Wait Resource	The resource for which the process is waiting.
Deadlock Priority	The deadlock priority specified for the SQL Server session.

Databases

The Databases pane displays all of the databases that were involved in deadlock situations during the specified time range. This pane contains the following sections:

- Involved Databases table — contains the following columns
  - Name — the name of the involved database.
  - Overall lost time — the total amount of time that the terminated sessions (“deadlock victims”) were running before they were terminated and rolled back.
  - Deadlocks — the total number of deadlock situations in which the database was involved during the specified time range.
  - Chained — indicates how many of the deadlock situations were chained deadlocks, that is, deadlocks that involve more than two sessions.
  - Regular — indicates how many of the deadlock situations involved only two sessions.
- Deadlocks related to a database — clicking a specific database in the Involved Databases table displays in this section data about the deadlocks in which the selected database was involved, using the following components:
  - Chart — displaying the distribution over the specified time range of deadlocks in which the selected database was involved.
  - Table — displaying detailed data about each of the deadlocks in which the selected database was involved. This table is identical to the one displayed on the Deadlocks pane. For details, see Deadlock table on page 95.

Objects

The Objects pane displays all of the objects that were involved in deadlock situations during the specified time range. This pane contains the following sections:

- Involved Objects table
- Deadlocks related to an object — clicking a specific object in the Involved Objects table displays in this section data about the deadlocks in which the selected object was involved.

The components of this pane (tables and a chart) are identical to the ones found in the Databases pane.

Applications

The Applications pane displays all of the applications that were involved in deadlock situations during the specified time range. This pane contains the following sections:

- Involved Applications table
- Deadlocks related to an application — clicking a specific application in the Involved Applications table displays in this section data about the deadlocks in which the selected application was involved.

The components of this pane (tables and a chart) are identical to the ones in the Databases pane.

Viewing I/O Statistics by Database Files

The **I/O by File** panel displays current I/O statistics for each SQL Server file. The I/O statistics' display can also be grouped by other grouping criteria, such as database files or disk.

This panel includes the following components:

- A table that displays current SQL Server I/O statistics for each SQL Server database file. For details, see Viewing the I/O by File table.
- A chart that provides a graphic representation of some of the I/O statistics for each row selected in the table. For details, see Viewing the I/O by File chart on page 97.

Viewing the I/O by File table

The *I/O by File* table displays current SQL Server I/O statistics for each SQL Server database file.

To create a custom filter for this table, use the options accessible by clicking the **Customizer** button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 33. The *I/O by File* table displays by default the following columns:

Column	Description
DBName	The name of the SQL Server database that contains the file.
File Name	The name of the file whose I/O statistics are on display.
Disk	The name of the physical disk on which the database resides.
MB on Disk	Physical file size on disk in megabytes.
MB Read	The number of megabytes read from this file since SQL Server started.
MB Written	The number of megabytes written to this file since SQL Server started.
MB Total	The total number of megabytes written to and read from this file since SQL Server started.
Reads Wait Time	The total number of seconds SQL Server has spent waiting for physical read operations on the file since SQL Server started. Displays data only for SQL Server 2005 and later versions.
Writes Wait Time	The total number of seconds SQL Server has spent waiting for write operations on the file since SQL Server started. Displays data only for SQL Server 2005 and later versions.
Total Wait Time	The total number of seconds SQL Server has spent waiting for physical read and write operations on the file since SQL Server started. Displays data only for SQL Server 2005 and later versions.
Read Operations	The total number of read operations carried out by this file during the selected time period.
Write Operations	The total number of write operations carried out by this file during the selected time period.
Total Operations	The combined total number of read and write operations carried out by this file during the selected time period.

Viewing the I/O by File chart

The *I/O by File* chart provides a graphical representation of SQL Server I/O statistics for each SQL Server database file.

This chart represents several indicators from the table (see Viewing the I/O by File table on page 97), as presented in the following table.

To display only requested files, click the first requested file and then hold down either the **Shift** key, for choosing a block of multiple files, or the **Ctrl** key, for choosing individual files.

Table 34. The I/O by File chart displays the following:

Indicator	Description
Current IO Rate	The current rate, in megabytes, at which SQL Server is performing physical read and write operations to this file (Reads rate plus Writes rate).
Current Read Rate	The current rate, in megabytes, at which SQL Server is performing physical read operations from this file.
Current Write Rate	The current rate, in megabytes, at which SQL Server is performing physical write operations to this file.
Current Wait Time (Average Active Sessions)	The number of seconds that SQL Server has spent waiting for I/O operations on this file since the last time data was collected.

Reviewing Database Usage

The Databases drilldown displays storage information about the selected SQL Server instance, including: databases, file groups, files, tables, indexes, disks, and log files.

The Databases drilldown is divided into the following areas:

- Database Details — contains the Summary pane, which provides a graphical representation of the database space and history (either for a single database or for multiple databases), as well as several panes that display lower levels of the storage hierarchy.
- TempDB — helps analyze TempDB activity and diagnose potential problems. For details, see Monitoring TempDB Status on page 112.

Monitoring SQL Server Databases

The *Databases* table provides details about all of the selected SQL Server databases within the selected instance. The information displayed in the table can be changed by selecting, from the View list, one of the options detailed in the following sections:

- Overview — general information about each database.
- Transactions on page 99 — details about the oldest active transaction in each database.
- Backup Status on page 100 — details on database backups, including date and total file sizes.
- Properties on page 100 — the properties of the various SQL Server databases.
- In Memory on page 100 -

Overview

To refresh

Table 35. The *Overview* view displays general information about each of the monitored databases.

Parameter	Description
dbid	ID of the database, unique within an instance of SQL Server.
Database Name	Name of the database, unique within an instance of SQL Server.
Status	The database's current status (for example, ONLINE, OFFLINE, RESTORING, or SUSPECT).
Database Space	Provides a visual representation of the database size and space usage within each database.

Table 35. The *Overview* view displays general information about each of the monitored databases.

Parameter	Description
Data Size	The total size allocated to data files.
Data Used Size	The total size actually used by data files in the database.
Data Free Size	The total available data size (allocated to data files but not used).
Log Size	The total size allocated to log files.
Log Used Size	The total size actually used by log files in the database.
Log Free Size	The total available log size (allocated to log files but not used).
# Tables	The number of tables, where table type=user, which exist in the database.
# Indexes	The number of indexes that exist in the database.
# File Groups	The number of file groups that exist in the database.
Last Backup Full Time	The date and time when the most recent full database backup of any type took place.
Last Backup Differential Time	The date and time when the most recent differential database backup of any type took place.
Last Backup Log Time	The date and time when the most recent database log backup of any type took place.
NOTE: The Last Backup Full Time, Last Backup Full Differential Time, and Last Backup Log Time columns are not displayed by default; to view these columns, click the **Customizer** button at the right side of the table	
Recovery Model	The recovery model for the database; for example, Simple, Bulk Logged, or Full.

Transactions

The Transactions view displays details about the oldest active transaction in each of the monitored databases. A transaction open for a long time may result in degraded performance.

Table 36. The *Transactions* view displays the following parameters:

Parameter	Description
dbid	ID of the database, unique within an instance of SQL Server.
Database Name	Name of the database, unique within an instance of SQL Server.
Oldest Tran Start Time	The date and time at which the oldest active transaction in the database began.
Oldest Tran SPID	The system process ID of the session that owns the oldest active transaction in the database.

To view the session details of a transaction:

1. Select the **SQL Activity** drilldown > **Sessions** panel > **Sessions** table.
2. In the Sessions table, browse for the requested SPID.
3. Click the SPID number to display the Session Details page.

   Use this page to review the information relevant for handling the transaction, such as the transaction count and the text of the most recent SQL batch.

Backup Status

The Backup Status view displays backup details for the database, such as the data size and the date and time of the most recent backup of any type for the database.

NOTE: The summarized value of the data size and the log size indicates the anticipated size of the backup file.

Table 37. The Backup Status view displays the following parameters:

Parameter	Description
dbid	ID of the database, unique within an instance of SQL Server.
Database Name	Name of the database, unique within an instance of SQL Server.
Status	The database's current status (for example, ONLINE, OFFLINE, RESTORING, or SUSPECT).
Data Size	The total size allocated to data files.
Log Size	The total size allocated to log files.
Last Backup Full Time	The date and time when the most recent full database backup of any type took place.
Last Backup Differential Time	The date and time when the most recent differential database backup of any type took place.
Last Backup Log Time	The date and time when the most recent database log backup of any type took place.

NOTE: The Last Backup Full, Differential, and Log Time columns are not displayed by default; to view these columns, click the Customizer button at the right side of the table.

Properties

The *Properties* view displays database properties that correspond to the data returned by the SQL Server DatabasePropertyEx function. For more information about these properties, see the Transact-SQL Reference section in Microsoft® SQL Server Books Online.

In Memory

The In Memory view displays metrics on space usage by objects in memory. To view details for a specific database, click on the database row. Detailed metrics will be displayed below the table.

Table 38. The In Memory view displays metrics on space usage by object in memory

Parameter	Description
dbid	The database ID number.
Database Name	The name of the database.
XTP Enabled	Whether or not the database is XTP enabled.
Resource Pool Name	The name of the resource pool allocated to the database.
Percent of Pool	Total memory usage of the objects within the database.
Tables Allocation	Total amount of memory (in MB) allocated for the tables.
Indexes Allocation	Total amount of memory (in MB) allocated for the indices.
System Allocation	Total amount of memory (in MB) allocated for the system.
Table Used	Total amount of memory (in MB) used by the tables, including row versions.
Indexes Used	Total amount of memory (in MB) used by the indices, including row versions.
System Used	Total amount of memory (in MB) used by the system, including row versions.

Monitoring Database Details

You can display detailed data for a database. by selecting the database row in any one of the tables displayed on the Overview, Transactions, Backup Status, Properties or In Memory tabs.

The Database Details panel allows carrying out the following tasks:

- Monitoring database space and history — using the **Summary** pane, which displays a graphical representation of the disk space usage for databases, as well as various statistics for each database. For further details, see Monitoring Database History on page 101.
- Monitoring file groups — using the **File Groups** pane; see Monitoring File Groups in the Selected Databases on page 102.
- Monitoring data files — using the **Data Files** pane, which displays all files (excluding the Transaction Log) in the selected databases; see Monitoring data files on page 103.
- Monitoring transaction logs — using the **Transaction Logs** pane; see Monitoring Transaction Logs in the Selected Database on page 104.
- Monitoring log files using the **Log Files** pane; see Monitoring Log Files on page 106.
- Monitoring tables — using the **Tables & Indexes** pane, which displays statistics for all tables and indexes in the selected databases (by default, 50 rows are displayed); see Monitoring Tables and Indexes on page 107.

  NOTE: The panes from *File Groups* to *Tables and Indexes* display tabular and graphical information about other areas down the storage level hierarchy, currently selected in the *Databases* table. The charts on each of these panes display the amount of space allocated to the area for each selected database (Space Chart), and the amount by which each of the areas is growing over time (Growth Chart).

- Monitoring disk space — using the **Disk Space** pane, which displays each logical disk on the server, with a detailed graphical representation of the disk space usage for all disk. For further details, see Monitoring Disk Space on page 111.

Monitoring Database History

The **Summary** pane displays three separate charts showing various statistics relating to recent I/O activity on each database. You can select which data each graph should display.

.

NOTE: The graphs in the chart are available for each database.

To view a particular graph, select it from the Database History list.

Table 39. The following graphs are available for each database:

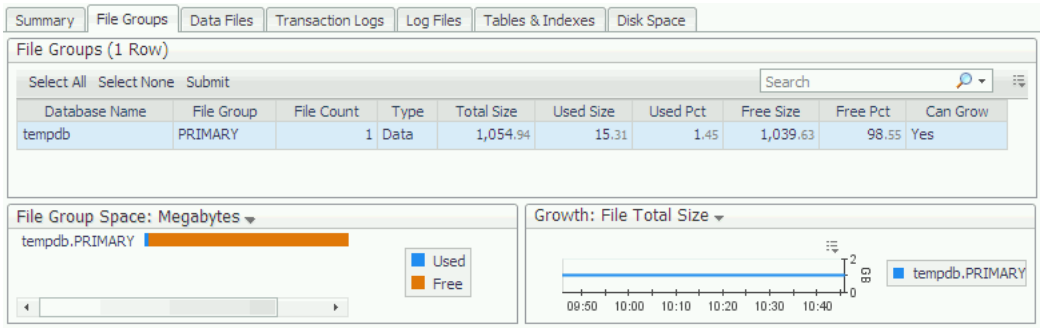
Graph	Description
% Log Used	The amount of file space allocated to the transaction log of each currently used database. Log space can be freed up by backing up the log, or truncating it, using the *truncate* option (`backup log <dbname> with truncate_only`). **NOTE:** All log records that exist after the oldest open transaction cannot be freed.
Data File Size	The amount of disk space that the data files are using. This graph indicates how the size of data files has changed over time.
Log File Size	The amount of disk space that the log files are using. This graph shows how the size of log files has changed over time.
Active Transactions	The number of open transactions in each database.
Transactions Rate	The rate of transactions for each database.

Table 39. The following graphs are available for each database:

Graph	Description
Log Cache Hit Rate	Percentage of log cache reads satisfied from the log cache. This metric reflects the amount of physical log I/O that is being avoided by caching log data. The log caches normally have a non-zero hit rate only if Rollback activity is taking place.
BCP Throughput Rate	The rate (measured in kilobytes per second) at which data is being loaded into the database using BCP (Bulk Copy Program) or BULK INSERT. Despite the speed at which BCP and BULK INSERT can import data into SQL Server, system I/O performance may degrade while BCP operations are underway.
Backup Throughput Rate	The rate (measured in kilobytes per second) at which backup or restore operations are reading or writing to the database. Throughput of a database backup or restore operation allows determining the progress and performance of these operations; for example, measuring how the performance of the database backup operation changes when more backup devices are used in parallel or when faster devices are used.
DBCC Scan Bytes Rates	The rate (measured in kilobytes per second) at which Database Console Commands (DBCC) are processing data. This rate represents the number of logical read scan kilobytes per second for database command console (DBCC) statements.
Log Flushes Rate	The rate at which the log cache for each database is being flushed to disk, which is necessary to guarantee that transactions can be recovered in the event of a system failure.
Log Flush Wait Time	The amount of time spent, in milliseconds, waiting for log flushes in each database. High log flush wait time can be caused by a slow or overworked disk subsystem. If a database has a consistently high Log Flush Wait Time that never changes, run the SQL command CHECKPOINT on that database to force another log flush and re-check the value in Foglight for SQL Server.
Log Growths Rate	The number of times the log has been expanded for each database.
Log Shrinks Rate	The number of times the log has been reduced for each database.
Log Truncations Rate	The frequency of log truncations for each database.

Monitoring File Groups in the Selected Databases

The **File Groups** pane displays detailed information about all file groups in the databases that are currently selected in the Databases table.



This pane includes the following areas:

- File Groups table
- File Group Space chart on page 103

- File Groups Growth chart on page 103

File Groups table

The *File Groups* table shows information about all file groups in the selected database. File stream types (used by memory optimized objects and file-stream objects) are now supported.

Table 40. This table displays the following parameters:

Column	Description
File Group	The name of the file group in the database.
File Count	The number of files in each file group.
Type	The type of file group; for example, data or log file.
Total Size	The total size of all files in the group.
Used Size	The total size of used space of all files in the group.
Used Pct	The percentage of used space of all files in the group.
Free Size	The total size of free space of all files in the group.
Free Pct	The percentage of free space of all files in the group.
Can Grow	Indicates whether the files in the file group (if any) can grow automatically as they fill.

File Group Space chart

The *File Groups Space* chart displays the amount of space allocated to each file group. Each file group consists of used space and free space.

The chart values can be displayed in megabytes or as a percentage of disk space. Use the File Groups Space list to change the values displayed on the chart.

File Groups Growth chart

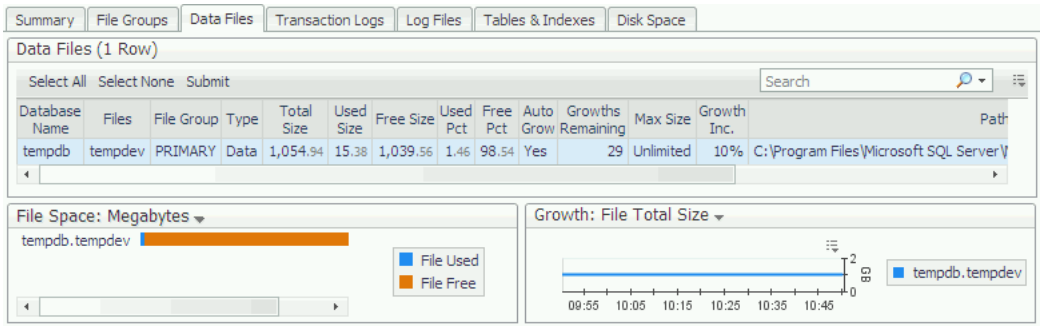
The *Growth* chart displays the amount by which each file group is growing over time.

Use the **Growth** list to change the values displayed on the chart. This list can be displayed by the total amount of space (File total size), used space (File used size), or unused space (File free size).

Monitoring data files

The ***Data Files*** pane displays detailed information about all data files in the database currently selected in the Databases table.

Figure 22. The *Data Files* pane.



This pane includes the following areas:

- Data Files table

- Data Files Space chart on page 104
- Data Files Growth chart on page 104

Data Files table

To create a custom filter for this table, use the options accessible by clicking the **Customizer** button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 41. The *Data Files* table displays the following indicators for all database data files:

Column	Description
Files	The name of the data file.
File Group	The name of the file group to which the file belongs.
Type	The type of file group; for example, data or log.
Total Size	The size of the file.
Used Size	The total size of the file's used space.
Free Size	The total size of the file's free space.
Used Pct	The percentage of the file's used space.
Free Pct	The percentage of the file's free space.
Auto Grow	Indicates whether the file can grow automatically.
Growths Remaining	The number of times the file can grow, considering its currently configured autogrow increment, before exceeding its space limits.
Max Size	The maximum size to which the file can grow.
Growth Inc.	Growth increment. Indicates the amount by which the file grows every time, if the *autogrow* option is enabled. for example, if the data file's initial size is 1 megabyte and it has to grow to a final size of 11 megabytes, setting the *Growth Inc.* parameter to 2 megabytes causes the file to grow five times, each time by a 2 megabyte increment.
Path	The path of the data file.
Db and File	The database name, concatenated with the database file name.

Data Files Space chart

The *File Space* chart displays the amount of space allocated to the data file for each selected database. Each data file consists of used and free space.

The chart values can be displayed in megabytes or as a percentage of disk space. Use the **File Space** list to change the values displayed on the chart.

Data Files Growth chart

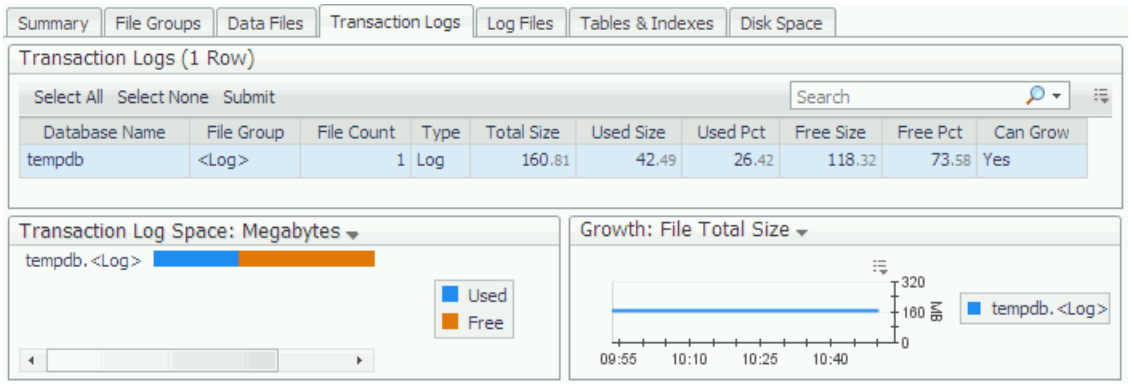
The *Data Files Growth* chart displays, in megabytes, the amount by which each file is growing over time.

Use the **Growth** list to change the values displayed on the chart. This list can be displayed by the total amount of space (File Total Size), used space (File Used Size) or unused space (File Free Size).

Monitoring Transaction Logs in the Selected Database

The ***Transaction Logs*** pane displays detailed information about all transaction logs in the database currently selected in the Databases table.

Figure 23. The *Transaction Logs* pane.



This pane includes the following areas:

- Transaction Logs table
- Transaction Log Space chart on page 105
- Transaction Log Growth chart on page 105

Transaction Logs table

The *Transaction Logs* table displays information about all transaction logs in the selected databases, as detailed in the following table.

Table 42. The *Transaction Logs* table details.

Column	Description
File Group	The name of the file group to which the file belongs.
File Count	The number of transaction logs in the database.
Type	The type of file; for example, log.
Total Size	The total size of all files in the transaction log.
Used Size	The amount of used space.
Used Pct	The percentage of used space within the log file.
Free Size	The amount of free space.
Free Pct	The percentage of space available for use within the log file.
Can Grow	Indicates whether any of the files in the transaction log can grow.

Transaction Log Space chart

The *Transaction Log Space* chart displays used space and free space for each of the selected databases' transaction logs.

The chart values can be displayed in megabytes or as a percentage of disk space. Use the **Transaction Log Space** list to change the values displayed on the chart.

Transaction Log Growth chart

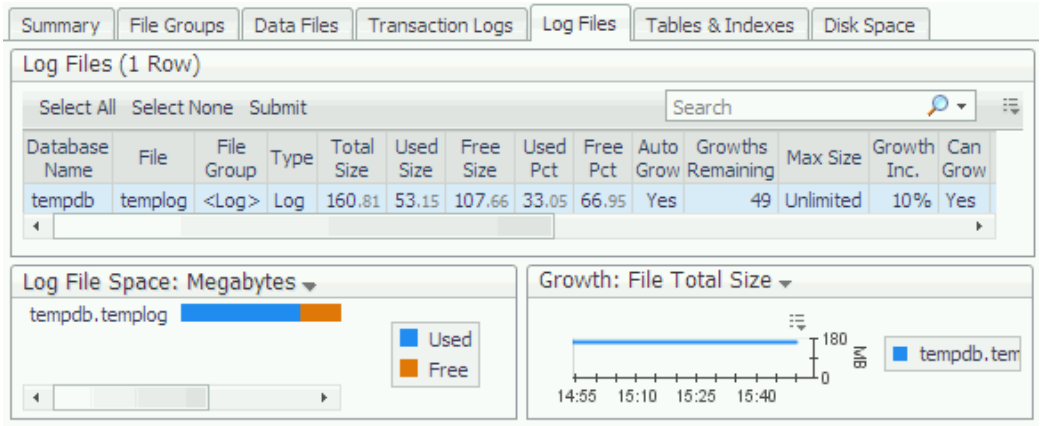
The *Transaction Log Growth* chart shows the amount by which each transaction log is growing over time.

Use the **Growth** list to change the values displayed on the chart. This list can be displayed by the total amount of either all transaction logs (File Total Size), used logs database (File Used Size), or unused transaction logs (File Free Size).

Monitoring Log Files

The ***Log Files*** pane displays detailed information about all log files in the database currently selected in the Databases table.

Figure 24. The *Log Files* pane.



This pane includes the following areas:

- Log Files table on page 106
- Log File Space chart on page 106
- Log File Growth chart on page 107

Log Files table

Table 43. The *Log Files* table displays information about all log files in the selected database.

Column	Description
File	The log file name.
File Group	The name of the file group to which the file belongs.
Type	The log file type.
Total Size	The size of the log file.
Used Size	The amount of the used space in the log file.
Free Size	The size of the currently unused space in the log file.
Used Pct	The percentage of used space within the log file.
Free Pct	The percentage of space available for use within the log file.
AutoGrow	Determines whether the log file can grow automatically.
Growths remaining	The number of times the file can AutoGrow before it can grow no more.
Max Size	The maximum size to which the log file can grow.
Growth Inc.	Growth increment. The amount by which the log file can AutoGrow at any one time.
Can Grow	Determines whether the log file can grow.
Path	The log file's path.
Db and File	The database name, concatenated with the log file name.

Log File Space chart

The *Log Files Space* chart displays the total space allocated to each log file.

The chart values can be displayed in megabytes or as a percentage of disk space. Use the **Log File Space** list to change the values displayed on the chart.

Log File Growth chart

The *Log File Growth* chart shows the amount by which each log file is growing over time.

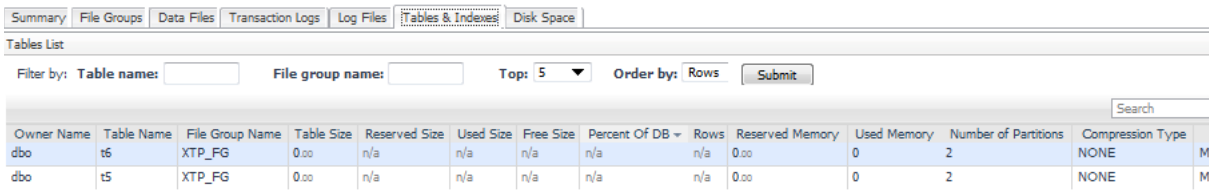
Use the **Growth** list to change the values displayed on the chart. This list can be displayed by the total amount of space (Total), used space (Used), or unused space (Free space).

Monitoring Tables and Indexes

The **Tables and Indexes** pane, which displays storage information for database tables and indexes, contains the following sections:

- Table list — provides detailed information about the disk space usage by the database's largest tables. For details, see Reviewing tables' space usage on page 107.
- Indexes list — displays all indexes created for the table selected on the Table list. For details, see Reviewing Indexes for the selected table on page 108.

Figure 25. The Tables and Indexes pane.



Reviewing tables' space usage

The Table list displays all tables in the selected database, including data such as their size, number of rows, free space, and owner name.

To set a user-defined display of the Table list:

1. Set the criterion for a large table — either size or number of rows — by selecting a value from the *Order By* list.
2. Set the number of tables to be displayed, by selecting a value from the *Top* list.

   Optionally — To filter the display, proceed to Step 4. To display the entire list according to the parameters set now, go to Step 5.
3. Use the *File Group Name* field to display only tables belonging to a specific file group.
4. Use the *Table* Name field to display either a specific table, by entering its exact name, or all tables that share a character string (for example, product), by entering that string.
5. Click **Submit**.

To create a custom filter for this table, use the options accessible by clicking the **Customizer** button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

The value displayed in the **Table Size** column indicates only the amount of space used by data, while the value displayed in the **Used Size** column indicates the amount of space used by both data and indexes, and the value displayed in the **Reserved Size** column indicates the entire amount of space reserved for data and indexes.

Table 44. The parameters displayed in the table are as follows:

Column	Description
Owner Name	The name of the table owner.
Table Name	The name of the table.
File Group Name	The name of the file group where the table is stored.
Table Size (MB)	The actual size of the table, in megabytes.
Reserved Size (MB)	The amount of space reserved for the table, in megabytes.
Used Size (MB)	The amount of used space in the table, in megabytes.
Free Size (MB)	The amount of free space in the table.
Percent of DB	The percentage of space that the table occupies in the database.
Rows	The number of rows in the table.
Reserved Memory	The amount of memory reserved for this table.
Used Memory	The amount of memory used by this table.
Number of Partitions	The number of partitions, if any.
Compression Type	What type of compression the table uses, if any.
Table Type	The type of table. Memory optimized, user tables and File tables are now included.

Reviewing Indexes for the selected table

The *Indexes* table shows the index statistics for all indexes in the selected tables. SQL Server stores statistics about the distribution of the key values in each index, and uses these statistics to determine which indexes to use in query processing.

Selecting a single row in the Indexes table displays detailed index statistics for that index in the lower section of the drilldown.

To create a custom filter for this table, use the options accessible by clicking the **Customizer** button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 45. The columns displayed in the Indexes table are as follows:

Column	Description
Owner Name	The owner of the index.
Table Name	The table that the index is associated with.
Index Name	The name of the index.
Index ID	The ID of the index.
File Group Name	The name of the File Group where the index resides.
Type	The type of the index.
No. of Keys	The number of keys in the index.
Index Size (MB)	The size of the index at the time of its last statistics update, measured in megabytes.
Used Size (MB)	The amount of used space in the index, in megabytes.
Free Size (MB)	The amount of free space in the index, in megabytes.
Rows	The number of rows in the table.

Table 45. The columns displayed in the Indexes table are as follows:

Column	Description
Row Mod Ctr	The number of row modifications made since statistics were last updated for this index.
Original Fill Factor	The fill factor used when creating an index, in order to reserve a percentage of free space on each data (leaf level) page.

This table allows selecting one or more indexes. When only one index is selected, its distribution statistics are displayed at the bottom part of the drilldown. When multiple indexes are selected, statistics are displayed only for the index with cursor focus (indicated by a black box around the index row).

Clicking a single index in the Indexes table displays the Index Properties pop-up.

The Index Properties pop-up contains the tables described in the following sections:

- Index Density table
- Fragmentation table on page 109
- Index Distribution chart on page 110

Index Density table

The *Index Density* table shows the density values for each combination of columns in the index.

Table 46. The Index Density table contains the following columns:

Statistic	Description
Columns	The name of the columns in the index.
Density	Density is the term used by SQL Server to represent the selectivity of the index columns. The more selective an index is, the more useful it is in searches. Density is calculated as 1 / distinct values for all values in the first key column of the statistics object, excluding the histogram boundary values. **NOTE:** Starting from SQL Server 2008, the density values is not used by the query optimizer, and is displayed only for backward compatibility with earlier versions.
Average Key Length	Average number of bytes per value for all of the key columns in the statistics object.

Fragmentation table

The *Fragmentation* table displays all indexes in the selected tables and shows the latest fragmentation information collected for each index.

Fragmentation information is collected by running the DBCC SHOWCONTIG command on each selected index.

NOTE: Running DBCC SHOWCONTIG on large tables can take a long time and put significant load on the server.

For information about the Fragmentation statistics displayed on the Fragmentation page, see the DBCC SHOWCONTIG topic in the Transact-SQL Reference section in the Microsoft® SQL Server Books Online.

How fragmented are my tables?

The following values can be used for determining the overall fragmentation in tables:

- *Extent Scan Fragmentation* and *Logical Scan Fragmentation* are expressed as percentage values.

  NOTE: A value of -1 is displayed for heaps (tables without indexes) and for text structures.

- *Average Page Density* shows how full each page is. This is affected by the FILL FACTOR setting used when creating the index; for example, immediately after creating an index with a FILL FACTOR of 30, the Average Page Density for that index is 30. This value is then change as data in the table is modified. Average Page Density can be thought of as the “real” fill factor at the time the data was collected, as opposed to the “original” fill factor that was specified when the index was created. The Original Fill Factor is used when an index is created to leave free space in index pages to allow for the insertion of new records without having to split the page.
- *Fill Factor Variance* is a computed metric, used for comparing the Average Page Density with the original fill factor used when creating the index. The value of this metric is calculated as the absolute difference between the average page density (AvgPage Density) and Original Fill Factor. The value of this variance shows how much the fill factor of the index has changed since the index was created.
- *Scan Density* is a value computed by dividing the optimal number of extent switches by the actual number of extent switches. It is expressed as a percentage value, where higher values indicate less fragmentation. Scan density is displayed with a value of -1 for tables that reside on more than one file.

Removing fragmentation

Several options are available for eliminating (or at least reducing) fragmentation. In general, fragmented indexes should be rebuilt by SQL Server. For details, see Microsoft® SQL Server Books Online.

Index Distribution chart

When a single index is selected in the Indexes table, the *Index Distribution* chart displays the index distribution histogram for the selected index.

When SQL Server collects statistics on an index for use in determining the indexes usefulness for resolving queries, it samples the data in the index and produces a histogram of the key values found. This chart shows that histogram for the selected index.

The values shown in the Index Distribution chart derive from the latest statistics for the selected index. If the index statistics are out of date, these values are also outdated. Check the Last Updated statistic on the Statistics page for the date when statistics were last collected. Use the **Update Statistics** SQL command to re-collect statistics.

Each series in the chart corresponds to one sample value from the first column in the index. The chart shows the number of rows in the table that have that value.

NOTE: SQL Server does not inspect every key value, but uses only sampling. The histogram displays the percentage of the table that falls in a sample range. Not all of the records counted have the exact sample value shown.

Example:

If the selected index is on a Name column, the chart may display the following values:

- ABRAHAM: 5
- GEORGE: 20
- PETER: 25
- STEVEN: 30
- ZACH: 20

This means that five of the people have a name that is alphabetically before or equal to ABRAHAM, 20 people are between ABRAHAM and GEORGE, 25 people are between GEORGE and PETER, and so on.

This chart can be used for identifying skewed indexes, that is, indexes that have a very uneven distribution of data. For example, if 95% of the table has the same value for a key, it is difficult for SQL Server to determine if it is useful index. Searches on values that fall in the 5% might find the index very useful, but searching on the value that occurs 95% of the time are not so useful.

Monitoring Disk Space

The **Disk Space** pane displays storage information about disks that reside on the currently monitored server.

The term disk in this tab is used on the assumption that each physical disk contains only one partition. If the disk is divided to several partitions (logical drives), the tab displays disk space utilization per partition, and not a summary of the utilization of the entire physical drive.

This pane is designed to answer the following questions:

- How much space are the SQL Server database files using on each disk?
- What file space is available?

This pane features the Disk Space chart, which displays each disk on the server. The display is divided into SQL Data files, SQL Log files, Non-SQL files or Free Space. This chart allows to easily view how much of each disk is used by SQL databases, and how much is free.

Table 47. The Disk Space chart displays several parameters:

Parameter	Description
Data Used	The amount of space on the disk used by data files in the currently monitored SQL Server databases.
Data Free	The amount of space on the disk allocated to data files in the currently monitored SQL Server databases but not used.
Log Used	The amount of space on the disk allocated to SQL Server log files for the currently monitored databases.
Log Free	The amount of space on the disk allocated to log files in the currently monitored SQL Server databases but not used.
Disk Free	The amount of space on the disk not used by any files.
Non SQL Files	The amount of space on the disk used by files not associated with the currently monitored SQL Server databases.

To include the *Disk Free* and the *Non SQL files* parameters on the chart, select the box *Include Non-SQL files in the chart*.

To restrict the data and log-related figures to display only the space used by specific databases, select one or more of the databases in the *Databases* table.

To choose whether to display the chart in megabytes (MB) or percentage, use the **Disk Space Utilization** drop-down list box.

NOTE: The space shown for SQL data and log files refers only to files in this instance of SQL Server. If multiple SQL Server instances are running on this server, the data/log files for other instances are included in the Non-SQL Files figure.

If the Windows server being monitored does not have Logical Disk performance counters enabled, the value of the Non-SQL Files component of this chart is always zero. If disk counters are not enabled, the Disk Counters Disabled alarm is displayed on the home page. To enable disk counters, it is advisable to run the *exctrlst.exe* utility, provided by Microsoft.

To use the exctrlst.exe utility:

1 Download the utility from the following location:

http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=7FF99683-B7EC-4DA6-92AB-793193604BA4

2 Double-click the *exctrlst.exe* binary file, to display the Extensible Counter List dialog box.

Use this dialog box to either enable the Performance Counter Resource indicated by the Collector log, or manually review every performance counter to ensure that its check box is selected.

Monitoring TempDB Status

The TempDB system database is a global resource that exists in any SQL Server instance and is used for various functions within it. There is one such database in every instance.

It is used to hold the following:

- User objects — Temporary objects explicitly created by users such as global or local temporary tables, temporary stored procedures, table variables, or cursors.
- Internal objects — Created by the SQL Server Database Engine. For example, work tables to store intermediate results for spools or sorting.
- Version store — Row versions that are either generated by data modification transactions in a database that uses read-committed using row versioning isolation, or snapshot isolation transactions, or Row versions that are generated by data modification transactions for features, such as: online index operations, Multiple Active Result Sets (MARS), and AFTER triggers.

Given its wide range of usage and the fact that there is only one such database in any instance, it may become a bottleneck. The TempDB dashboard aim is to help analyze TempDB activity and diagnose potential problems.

Monitoring TempDB Space Usage

The upper pane of the dashboard indicates the amount of space currently allocated across all data files of the TempDB database as well as its log files.

The usage profile graph indicates how space is used within TempDB so it is possible to see which component is allocated the majority of the space: user objects, internal objects or the version store.

The usage profile tab on the lower pane of the screen, provides more information about the usage trends of TempDB. The left side indicates how space is allocated between the various consumers of TempDB both at the database level and at the file level. To find out which tables are currently allocated within TempDB use the T*ables & Indexes* tab.

The Right side provides more information about the version store usage.If you see the version store consume a considerable amount of space in the TempDB, check the generation rate compared to the cleanup rate. A high generation rate that is not matched by the cleanup rate may indicate data modification transactions are not committing frequently enough. The "Current Transaction Running Time" indicates the running time of the longest transaction.

To find out more about sessions that are currently allocating space in TempDB, use the Sessions tab. The TempDB Current Allocated Size metric indicate the amount of space (MB) currently allocated by the session and not yet deallocated. The TempDB Total Allocated Size metric indicates the amount of space (MB) that the session has used since it started.

More details about the session can be found by drilling down to the Session dashboard using the link on the SPID column.

Monitor TempDB Activity

The Create Objects Rate graph, on the upper pane of the dashboard, can be used to determine the trend of user objects creation rate. If this number is high or deviates from its normal behavior it may indicate misuse of temporary objects.

The Summary tab, on the lower pane, can be used to examine rate of transactions against TempDB.

Wait for IO on TempDB files, as well as contention for allocation pages, can be examined in the SQL Performance dashboard if SQL PI is installed

Reviewing the Services

The Services drilldown provides detailed information, represented by graphs and tables, of the state of the various SQL Server support services.

Reviewing the Support Service Status

The **Services Status** panel includes panes that allow carrying out the following tasks:

- Monitoring the Status of the Current Services — using the Services Status table, which allows viewing the current status of each SQL Server support service, as well as the exact time when this status was last changed.
- Tracking the Service Status History on page 113 — using the Service Status History chart, which displays the status of the support services over the last hour.

Monitoring the Status of the Current Services

The Services Status table, which displays the current status of each SQL Server support service, contains several parameters, as presented in the following table.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 48. The Services Status table parameters.

Column	Description
State	The state of the SQL Server support service. The possible values are: Stopped, Paused, Running, Not Installed, Not Configured, and Configured.
Service	The SQL Server service name. The support services monitored by Foglight for SQL Server are as follows: • SQL Server Agent • DTC: This is only supported for SQL Servers running on Windows plaftorm. • Full Text Search • OLAP Services • SQL Server Mail • Integration Service • Report Service • Browser Services • Writer Services • ADH Service

Tracking the Service Status History

The Service Status History chart displays the status of each SQL Server support service over time. This table allows viewing whether a service has been installed, and the exact times when each support service's state was one of the following:

- Stopped
- Paused
- Running
- Not installed
- Not configured
- Configured

NOTE: *SQL Server Mail* and *SQL Agent Mail* are not services and cannot be started or stopped. The Service Status History chart displays only their current configuration status.

Reviewing SQL Agent Jobs

The **SQL Agent Jobs** panel lists all of the currently defined SQL Agent jobs, highlighting their current status (running, succeeded, or failed). This panel also allows viewing the execution messages from the last run of each job.

In addition, the SQL Agent Jobs panel displays the status of each job over time, thereby providing the exact information about when a job ran and when it succeeded or failed.

This panel allows carrying out tasks described in the following topics:

- Tracking the SQL Agent Jobs on page 114
- Tracking Job Execution Messages on page 115
- Viewing the SQL Agent Jobs History on page 115

Tracking the SQL Agent Jobs

The *SQL Agent Jobs* table lists all SQL Server Agent Jobs defined in this server. This table highlights the current status of each job (running, succeeded, or failed). This data is updated on demand.

IMPORTANT: To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 49. The job state is indicated to the left of the job name.

Indication	Description
	The job was never run.
The following indicators refer to jobs that were run:	
Blue	The job run was cancelled.
Red	The job run failed.
Green	The job run completed successfully.

The job state depends on the last run outcome. For example: if the current status is *Completed*, but the last run outcome is *Fail*, the job is indicated by a red circle as a failed job.

Table 50. The SQL Agent Jobs table contains the following columns:

Column	Description
Job Name	The name of the job.
Job Category	The SQL Agent job category assigned to this job.
Enabled	Indicates whether the job is enabled.
Last Run Outcome	The result of the last run (*Fail*, *Success*, *Retry*, *Cancel*, or *In progress*).
Current Status	The current status of the job (*Running*, *Completed*, or *Never Ran*).
Curr Step #	If the job is currently running, displays the current step number.
Last Run Time	The date/time on which the job last ran.
Last Run Finish	The time the job was completed.
Last Run Duration	How long the last run took (displayed in the *d hh:mm:ss* format).
Next Run Time	The date/time of the next scheduled run.
Description	A brief description of the job.

Tracking Job Execution Messages

The *Job Messages* table displays the execution messages for each step in the current job. The data in the table is refreshed on demand. Historical data is not stored in a repository.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 51. The Job Messages table displays the following parameters:

Column	Description
Time	The time the job or step started execution. For an In Progress history, this is the time the history was written.
Message	Displays an error message, if the current step in the job resulted in an error.
Step Name	The name of the step.
Run Status	The job execution's status (Failed, Success, Retry, Canceled, or In progress).
Run Duration	Elapsed time in the execution of the job or step in *seconds* format.

Viewing the SQL Agent Jobs History

The *SQL Agent Jobs History* chart displays the status of jobs over time, thereby providing the exact information about when a job ran and when it succeeded or failed. The possible statuses are as follows:

- Never Run
- Retrying
- Running
- Failed
- Canceled
- Success

When a job executes, Foglight for SQL Server adds it to this chart (Running), identifying the time it started.

When the job finishes, Foglight for SQL Server checks the completion status (Success or Failed) and changes the color of the job accordingly, to indicate when exactly a job started and ended.

Viewing SQL Agent Alerts

The **SQL Agent Alerts** panel lists all currently defined SQL Server Agent Alerts, highlighting the alert type and when it last occurred.

> **NOTE:** This **SQL Agent Alerts** panel is only supported for monitored SQL Servers running on Windows platform.

This panel also features a chart that allows tracking the occurrences of each alert over time.

The sections of this panel allow carrying out the tasks detailed in the following topics:

- Tracking SQL Agent Alerts
- Tracking Recent Alert Occurrences on page 116

Tracking SQL Agent Alerts

The *SQL Agent Alerts* table lists all SQL Server Agent alerts defined in this SQL Server instance.

This table highlights the type of alert (either Event Alert or Performance Alert), as well as how often the event has occurred, and the date and time of its last occurrence.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 52. The SQL Agent Alerts table displays the following parameters:

Column	Description
Alert Name	The name of the alert that was raised.
Type	The type of alert.
Enabled	Indicates whether the alert is enabled.
Last Occurred	The time when the alert was last raised.
Count	The number of occurrences of the alert.
DBName	The name of the database where the alert was raised.
Description	The full text of the specified alert's error message. This text is taken either from SQL Server or, for user-defined alerts, from the user-created message.

Tracking Recent Alert Occurrences

The *Alert Occurrences* chart shows when recent SQL agent alerts have occurred, and how many have occurred during the selected time range. The legend on the right indicates the types of alerts currently displayed in the chart.

Monitoring SQL Server Transactions using the DTC Panel

The **DTC** panel provides additional data on SQL Server transactions.

DTC (Distributed Transaction Coordinator) service is an integral component of Microsoft Windows (Windows 2003 and higher versions) and is also available as part of the SQL Server installation, if a previous version of Windows is used. For details, see glossary definition of DTC on page 290.

If the server to which Foglight for SQL Server is currently connected has more than one instance running, this panel provides data regarding all instances of SQL Server on the current computer, not only the instance currently under analysis.

Table 53. The DTC Details chart displays data on the following aspects of DTC performance:

Data Series	Description
Aborted Transactions	The number of distributed transactions that were rolled back.
Active Transactions	The number of currently active distributed transactions.
Committed Transactions	The number of distributed transactions committed per second.
In Doubt Transactions	Transactions that have passed phase 1 of the two-phase commit operation (have committed the local transaction), and are awaiting a response from the DTC to either commit or roll back (phase 2).
Response Time Average	The average of the full time (in milliseconds) it has taken a query (select 1, by default) to get from the application to SQL Server and back.
Transactions/sec	The number of distributed transactions performed per second.

Monitoring Full-text Indexes using the Full Text Search Panel

In full-text indexing, a separate catalog is maintained that indexes each word in a database field as a separate index entry. The **Full Text Search** panel displays performance details for all full-text indexes on the server.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 54. The *Full Text Search Service* table contains the following data.

Data Series	Description
Database Name	The name of the database where the tables in the Full Text Catalog reside. Provides a unique identification of the database within a SQL Server instance.
ID	ID of the full-text catalog. Provides a unique identification across the full-text catalogs in the database.
FTC Name	The name of the full-text catalog. Provides a unique identification within the database.
Status	The status of the full-text catalog. Can have one of the following values: • 0 — Idle • 1 — Full population in progress • 2 — Paused • 3 — Throttled • 4 — Recovering • 5 — Shutdown • 6 — Incremental population in progress • 7 — Building index • 8 — Disk is full. Paused. • 9 — Change tracking
Index Size MB	The size of the full-text index, in megabytes, rounded to the nearest value.
Item Count	The number of items currently found in the full-text catalog.
Path	The name of the catalog directory in the file system.
Last Populated	Displays the time when the full text index was last populated; returns 0 if no population has occurred.
Table Count	The number of tables in the database that take part in the full text indexing process.

Using the HADR Drilldown

The HADR drilldown provides monitoring information regarding the disaster recovering solutions provided by SQL Server: Log Shipping, Cluster, Mirroring, and Always On.

While the data in the Log Shipping panel is available for all versions of SQL Server:

- The Cluster panel data is only available for instances running on a cluster server that is installed on Windows platform.
- The Mirroring panel data is only available for instances that are running on Windows platform and have databases configured for database mirroring.
- The Always On panel requires a SQL Server version equipped with the Always On capability (SQL Server 2012 and higher).

Monitoring the Log Shipping

Log shipping is used for setting up backup databases to take the place of a current “live” database if that database goes down. To that end, database logs that are dumped on the live database are copied to the backup server and restored on the backup database.

Monitoring log shipping requires connecting to a monitor server (which may be the live server, one of its backups, or an outside monitor that does not take place in the log shipping process). The monitoring server is used for analyzing log shipping performance.

Foglight for SQL Server monitors log shipping for the following types of errors:

- Backup threshold exceeded — the failure to back up a database log within a set period of time.
- Out of Sync threshold exceeded — the failure to restore a database log to a backup server within a set period of time after the log has been backed up.

  For details, see Log Shipping Alarm on page 171.

The Log Shipping panel allows investigating the cause for a log shipping alarm by carrying out the tasks detailed in the following topics:

- Tracking the Performance of Servers used in Log Shipping
- Viewing the Log Shipping Details on page 119.

Tracking the Performance of Servers used in Log Shipping

The *Log Shipping* table contains performance details for the primary and backup servers used in Log Shipping, as presented in the table below.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 55. The Log Shipping table contains performance details.

Column	Description
Server	The server name (Primary server for the parent instance, and Secondary server for all subsequent child instances).
DBName	Depending on the instance role: • For the primary instance, the backup database on the source server • For the secondary instance, the destination database on the secondary server
Last Activity	Depending on the instance role: • For the primary instance, the last time the database was backed up • For the secondary instance, the last time a file was copied or restored
Activity Type	What activity was performed at the Last Activity time (Backup or Copy).
Threshold	The maximum allowed time (in minutes) before a Log Shipping alert occurs. The amount of time depends on the instance role: • For the primary instance, this is the maximum allowed time since the last transaction log backup was made on the source server. • For the secondary instance — the maximum allowed time between the last backup on the source server and the last restore on the secondary server.
Outage	During an outage, no alerts occur when thresholds are exceeded (for both the parent and child instances). If this field is blank, no outage takes place.
Threshold Alert	The Error Log drilldown displays this alert if the threshold is exceeded.
Alert Enabled	If this option is selected, only enabled alerts are displayed.

Viewing the Log Shipping Details

Table 56. The Log Shipping Details section displays the following indicators about the currently selected activity:

Column	Description
Last Backup File	The name of the most recent backup file.
Last File Copied	The name of the most recent backup file that was copied to the secondary server.
Last File Restored	The name of the backup file that was most recently restored to the secondary server.
Backup Time	The time when the most recent backup was carried out.
Copy Time	The date the backup file was most recently copied from the primary to the secondary server.
Restore Time	The date the backup file was most recently restored to the secondary server.
Under the **Current Time** section:	
Source Server	The current timestamp at the primary server.
Target Server	The current timestamp at the secondary server.
Monitor Server	The current timestamp at the monitor server.

Monitoring Cluster Services

The **Cluster Services** panel displays information about the state of the current Microsoft Cluster Server.

NOTE: If the currently connected SQL Server instance is not part of a Microsoft Cluster Server, this page displays the message *SQL Server is not running on a Cluster server*.

This panel is used for investigating the causes for the Cluster Server Down alarm, which is raised when Foglight for SQL Server detects that at least one cluster node (server) is not currently running as part of the cluster.

The Cluster Services panel provides only tabular information, using the Cluster Services table. For details, see Monitoring the State of the Current Microsoft Cluster Server on page 119.

Monitoring the State of the Current Microsoft Cluster Server

The Cluster Services table displays information about the state of the currently monitored Microsoft Cluster Server.

This table allows viewing the status of each cluster resource and group, as well as the status of any cluster resources owned by each server (node) in the cluster. Foglight for SQL Server highlights any unusual conditions such as resources offline, or cluster nodes down.

To create a custom filter for this table, use the options accessible by clicking **Advanced**. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 57. The Cluster Services table displays indicators of the Cluster Server performance:

Column	Description
Name	The hierarchical tree of cluster resources, the root of which is the name of the cluster. Lower levels in the tree represent cluster groups, resource groups and servers, and resource details.
Status	The status of the current resource (where applicable).
Server	The individual server in the cluster where the specified resource is located.
Comment	A brief description of the specified cluster resource (if available).

Tracking the Status of the Mirroring Operation

If one or more databases within the monitored instance take part in a mirroring operation, either as a principal database whose exact copy is mirrored on a different instance, or as a mirror database, the Mirroring panel allows viewing the status of the mirroring operation.

If no database within the monitored instance takes part in a mirroring operation, the Mirroring panel is left blank and displays the note *This instance has no database configured for database mirroring*.

If the monitored instance supports the Always On feature (supported only for SQL Server version 2012 Enterprise edition), Always on cannot function at the same time with Mirroring operation: enabling Mirroring requires disabling Always on and conversely. Therefore, if the monitored instance contains databases that have Always On enabled, the Mirroring panel displays the note *This instance has no database configured for database mirroring.*

The Mirroring panel includes the following sections:

- Mirroring table — see Mirroring Table on page 120.
- Role and data flow of the selected database — see Viewing the Role and Data Flow of the Selected Database on page 122.

In addition, this panel allows further investigation of the selected database, by drilling down to other locations.

- Investigating using the Databases drilldown — carried out by clicking the instance name under the Database column.
- Viewing the database's mirroring performance history — carried out by clicking the instance's mirroring role (MIRROR or PRINCIPAL) under the Mirroring Role column. For details, see Viewing the Selected Database's Mirroring Performance History on page 123.
- Investigating using the partner's mirroring page — carried out by clicking the link at the upper right side of the screen or the Mirror database icon under the section Role and Data Flow of the Selected Database. For details, see Viewing the Partner's Mirroring Page on page 124.

Mirroring Table

The Mirroring table displays the columns listed below.

The table below provides data about the columns that appear by default in the table. For a list of additional metrics, which are hidden by default, see Additional metrics in the Mirroring table on page 122.

Table 58. Mirroring table, default columns:

Column	Description
DBID	A number (ID) uniquely identifying a database within a SQL Server.
Database	The database name. **NOTE:** The database name appears as a link which, when clicked, displays the Databases drilldown in Overview mode, with the selected database highlighted in the Databases table and its details displayed in the panes below.
Mirroring Role	The role the database takes in the mirroring process; either *principal* or *mirror*. **NOTE:** The mirroring role is displayed as a link which, when clicked, displays the mirroring performance history of the selected database. For details, see Viewing the Selected Database's Mirroring Performance History on page 123.
Principal	The name of the instance whose role in the process is *principal*.
Mirror	The name of the instance whose role in the mirroring process is *mirror*.

Table 58. Mirroring table, default columns:

Column	Description
Mirroring Status	Indicates the severity determined based on the database state: Normal, Warning, or Critical. Each of these severity level values can correspond to several mirroring states. *Normal* — can correspond to one of the following mirroring states: • SYNCHRONIZED (with a Witness) • SYNCHRONIZED (without a Witness) **NOTE:** To view whether the mirroring operation is configured to use a Witness, click the Customizer button at the end of the table and turn on the display of the Witness Name and Witness State columns. For details, see Additional metrics in the Mirroring table on page 122. • SYNCHRONIZING *Warning* — can correspond to one of the following mirroring states: • DISCONNECTED • PENDING_FAILOVER *Critical* — can correspond to one of the following mirroring states: • SUSPENDED • UNSYNCHRONIZED
Mirroring State	A state indicating the mirroring session condition. This field can have one of the following values: • DISCONNECTED • SYNCHRONIZED (with a Witness configured) • SYNCHRONIZING • PENDING_FAILOVER • SUSPENDED • UNSYNCHRONIZED • SYNCHRONIZED (without a Witness configured)
Safety Level	The safety level at which the mirroring session is configured to work. This column indicates the level of synchronization between the two servers that take part in the mirroring process. This field can have one of the following values: • UNKNOWN — unknown state • OFF — high availability (asynchronous). When an instance is configured with this safety level, the main consideration is that it remains available, regardless of the availability of its mirroring partner. • FULL — high safety (synchronous). The instance is configured to fail over when its mirroring partner becomes unavailable. Failover can be carried out automatically (if a Witness is configured) or manually.
Redo Queue	The redo queue size. This size can be either left Unlimited or defined in megabytes (MB).
Alarms	The number of warning, critical, and fatal alarms for the SQL Server database instance. When holding the cursor over one of the alarm counts, the dwell displays the most recent alarms raised against this database, sorted by severity. Clicking this field displays the Alarms list, which is listed by severity order. See the *Foglight Online Help, Monitoring System-Wide Alarms* for details on the alarm information.

Additional metrics in the Mirroring table

The metrics listed below, which are also part of the Database Mirroring collection, do not appear by default in the Mirroring table. For details on these metrics, see the *Database Mirroring Collection* section in *Foglight for SQL Server Reference Guide*.

To display the metrics listed below:

1 Click the **Customizer** button (

) at the end of the table.

2 Select the check box near the requested metric.

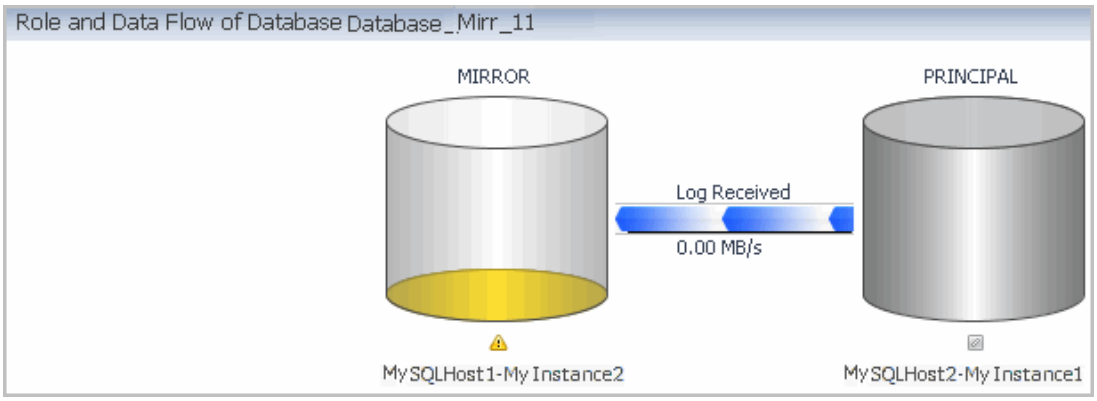
Additional metrics list

- Failover LSN
- End Log LSN
- Replication LSN
- Witness State
- Connection Timeout
- Partner Name
- Witness Name
- Role Sequence
- Log Scanned For Undo
- Log Remaining For Undo
- Mirroring Roundtrip - Latency
- Commit Acknowledgement Delay
- Write Commit
- Log Sent
- Log Send Flow Buffer Wait
- Log Sent From Cache
- Send Queue
- Roll Forward Queue
- Log Received
- Log Rolled Forward
- Log Cache Redone
- Log Harden Wait Time

Viewing the Role and Data Flow of the Selected Database

The *Role and Data Flow of Database* section displays the mirroring operation of the database selected in the Mirroring table and its mirroring partner. The monitored instance is always displayed on the left, and the data is shown as flowing in this instance's direction, that is: Log Received if the database is Mirror, and Log Sent if the database is Principal.

Figure 26. The Role and Data Flow of Database displays the mirrored databases.



To investigate the data flow by displaying the selected database's mirroring partner, use one of the following methods:

- Click the cylinder icon that represents the mirroring partner
- Click the link **Investigate using the Partner's Mirroring Page** on the upper right side of the panel, above the table.

If the partner is currently monitored, its mirroring page will now be displayed. For details, see Viewing the Partner's Mirroring Page on page 124. Otherwise, an error message is displayed, notifying that the partner server is currently not monitored.

Viewing the Selected Database's Mirroring Performance History

Clicking the Mirroring Role column of the Mirroring table displays the database's Mirroring Performance History page, which allows carrying out the tasks described in the following sections:

- Tracking the mirroring role over time
- Tracking data transfer between the principal and the mirror databases
- Tracking the mirroring roundtrip during the selected time range on page 124

Tracking the mirroring role over time

The Mirroring Role pane displays the role the database played in the mirroring operation during the selected time range: Principal, Mirror, or Not Mirroring.

The mirroring role is displayed as a gauge, with each of the mirroring roles indicated by another color. Positioning the cursor over the gauge displays the percentage, within the specified time range, the database spent in each role.

Tracking data transfer between the principal and the mirror databases

The middle section of the Mirroring Performance History page includes the following panes:

- **Principal Counters** — this pane displays a chart of the following values, which can be selected from the list on the upper left:
  - Commit Acknowledgement Delay — indicates a delay in waiting for acknowledgement of unterminated transaction commit. This metric is specific to the principal database, and holds values only when a Full safety level is configured.
  - Write Commit — the number of transactions in the principal database that had to wait for a write commit in the mirror database's transaction log. Low values of this metric indicate a bottleneck.
  - Log Sent — indicates the rate, in megabytes, of log sent from the principal to the mirror database.

    - Log Send Flow Buffer Wait — the amount of time the mirroring session had to wait to use the mirroring flow control buffer. This value is specific to the principal database.
    - Log Sent from Cache — indicates the aggregated size, in megabytes, of the log records sent from the principal database cache rather than straight from the transaction log.
    - Send Queue — total size, in megabytes, of data waiting to be sent to the mirroring database.
- **Mirror Counters** — this pane displays a chart of the following values, which can be selected from the list on the upper left:
    - Roll Forward Queue — total size, in megabytes, that remains to roll forward to the mirroring database.
    - Log Received — indicates the aggregated size, in megabytes, of log records received from the principal database.
    - Log Rolled Forward — the aggregated size, in megabytes, of log records that were rolled forward on the mirror database.
    - Log Cache Redone — the aggregated size, in megabytes, of log records that are being read from redo cache rather than from the transaction log. Constantly low values of this metric indicate that the transaction logs are arriving faster than they can be read by the redo task. This metric is specific to the mirror database.
    - Log Harden Wait Time — the amount of time spent waiting for the log to be written to the mirroring database. High values of this metric can indicate that the disk of the mirroring database is loaded.

Tracking the mirroring roundtrip during the selected time range

The Mirroring Roundtrip section displays a chart that indicates the latency of the mirroring session during the selected time range.

Viewing the Partner's Mirroring Page

When viewing the mirroring performance of a database that takes part in the mirroring operation, either as a principal or as a mirror database, it is possible to investigate the selected database's mirroring partner, using one of the following methods:

- Clicking the cylinder icon that represents the mirroring partner
- Clicking the link **Investigate using the Partner's Mirroring Page** on the upper right side of the table.

In so doing, the mirroring operation can be investigated from the point of view of the other database, thereby determining the source of the performance issue.

Reviewing the Always On Availability Groups

If one or more databases within the monitored instance is of version SQL Server 2012 Enterprise Edition, is part of an OS cluster and has the Always On feature enabled, the Always On panel allows viewing the status of the Always On availability groups that reside on the cluster.

NOTE: Always on cannot function at the same time with Mirroring operation: enabling Mirroring requires disabling Always on and conversely. Therefore, if the monitored instance contains databases that support Always On but have Mirroring enabled, the Always On panel displays the note *This instance has no database configured for Always On.*

The Always On panel includes the following sections:

- OS Cluster information — see OS Cluster Information on page 125.
- High Availability groups table — displays all High Availability groups that are configured on the instance. For details, see High Availability groups table on page 125.

- Availability group map — displays the dependency (primary/secondary relations) between the databases in the groups selected in the High Availability groups table, or between all groups in the table. For details, see Availability group map on page 128.

OS Cluster Information

The OS cluster information section displays the following details:

- OS Cluster Name — the name of the OS cluster hosting the SQL Server instances that are enabled for AlwaysOn Availability Groups.
- Quorum Type — the type of quorum used by the selected OS cluster; can have one of the following values:
  - 0 — Node Majority.

    The cluster is defined as healthy only if indicated so by more than 50% of the voting nodes in the cluster. For example, on a nine node cluster this quorum configuration can sustain up to four node failures.
  - 1 — Node and Disk Majority.

    This mode is similar to Node Majority quorum mode, with the addition of a shared disk cluster resource as a voting witness. More than 50% of the cluster's possible votes must be affirmative in order to determine the cluster's state as healthy, and connectivity from any node to that shared disk is also counted as an affirmative vote.
  - 2 — Node and File Share Majority.

    This mode is similar to Node Majority quorum mode, with the addition of a remote file share configured as a voting witness. More than 50% of the cluster's possible votes must be affirmative in order to determine the cluster's state as healthy, and connectivity from any node to that remote file share is also counted as an affirmative vote.
  - 3 — No Majority: Disk Only.

    A shared disk cluster resource is designated as a witness, and connectivity by any node to that shared disk is counted as an affirmative vote.
- Quorum State — the state of the quorum used by the selected OS cluster; can have one of the following values:
  - 0 — Unknown quorum state
  - 1 — Normal quorum
  - 2 — Forced quorum

High Availability groups table

The High Availability group table displays information about the availability and health of all availability groups configured in the instance and the databases they serve.

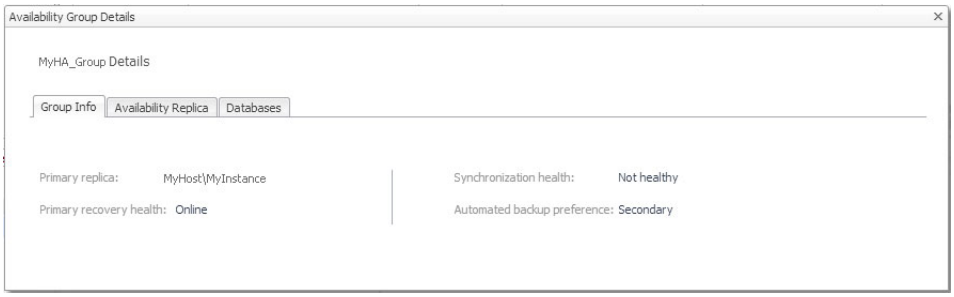
IMPORTANT: It is possible that one or more the availability replicas that take part in a group are not currently monitored. In such a case an indication appears near the replica in the availability group map. Click this indication to launch the instance monitoring wizard. For details, see Availability group map on page 128.

The High Availability group table includes the following columns:

- HA Group Status — displays an icon that indicates the worst severity reported within the group from the group alarms.
- Group Name — displays the High Availability group name.
- Role — the current state of the database that runs on the selected instance and belongs to the selected group; either Secondary or Primary. When there is no value for Secondary/Primary, this column should indicate the status *Off line*.
- Alarms — displays alarms related to the health of the availability group or its associated databases.

- Max Estimated Recovery Time — displays the maximum value of the estimated recovery time parameter within all databases that belong to the selected group. If all databases are fully synchronized, this column displays a zero value.
- Max Estimated Data Loss — displays the maximum value of the estimated data loss parameter within all databases that belong to the selected group. If all databases are fully synchronized, this column displays a zero value.
- Explore — contains a shortcut icon which, when clicked, displays the *Availability Group details* pop-up.

Figure 27. Availability Group details pop-up.



This pop-up contains the Group Info, Availability Replica, and Databases tabs.

The Group Info tab includes the following fields:

- Primary replica — the primary replica's name.
- Primary recovery health — indicates the primary replica's recovery health, with the following possible values: In progress, Online or Null.
- Synchronization health — summarizes the synchronization status of all replicas in the availability group:
  - Healthy — if all replicas are fully synchronized.
  - Partially healthy — if only some of the replicas are fully synchronized.
  - Not healthy — if no replica is fully synchronized.
- Automated backup preference — indicates whether backups should run on the primary replica. The possible values are:
  - 0 - Primary — backups should always be carried out on the primary replica.
  - 1 - Secondary_only — performing backups on a secondary replica is preferable.
  - 2 - Secondary — performing backups on a secondary replica preferable, but performing backups on the primary replica is acceptable if no secondary replica is available for backup operations. This is the default behavior.
  - 3 - None — no preference about whether backups are performed on the primary replica or on a secondary replica.

The Availability Replica tab lists all replicas associated with the group. This tab includes the following fields:

- Server Name — the name of the server.
- Join State — can have the following values:
  - Not joined
  - Joined standalone — joined as a standalone instance
  - Joined Failover — joined as failover cluster instance
- Role — either Primary or Secondary.
- Operational State — indicates whether the replica is online, offline or in a pending stage. The possible values are as follows:

    - Pending failover
    - Pending
    - Online
    - Offline
    - Failed, no quorum
    - Null — replica is not local
- Connected State — indicates whether a secondary replica is currently connected to the primary replica. Possible values are either Connected or Disconnected.
- Recovery Health — indicates whether all databases joined to the availability group on the availability replica, are online, or are being recovered after a failover. Possible values are:
    - Online
    - In progress — recovering from a failover
- Synchronization Health — summarizes the synchronization status of all replicas in the availability group:
    - Healthy — if all replicas are fully synchronized.
    - Partially healthy — if only some of the replicas are fully synchronized.
    - Not healthy — if no replica is fully synchronized.
- Secondary Role Allow Connections — indicates whether an availability replica that is performing the secondary role (that is, a secondary replica) can accept connections from clients:
    - No — no connections are allowed to the databases in the secondary replica, and the databases are not available for read access. This is the default setting.
    - Read only — read-only connections have exclusive access to the databases in the secondary replica. All databases in the replica are available for read access.
    - All — all connections are allowed to the databases in the secondary replica for read-only access.
- Availability Mode — can have either of the following values:
    - Asynchronous commit — the primary replica can commit transactions without waiting for the secondary to write the log to disk.
    - Synchronous commit — the primary replica waits to commit a given transaction until the secondary replica has written the transaction to disk.
- Failover Mode — either Manual or Automatic.

The Databases tab displays all databases associated with the group. This tab includes the following fields:

- Replica Server Name — the name of the server.
- Database Name — the name of the database.
- Database State — indicates whether the database is online, offline or in a pending stage. The possible values are as follows:
    - Online
    - Restoring
    - Recovering
    - Recovery pending
    - Suspect
    - Emergency
    - Offline
- Synchronization State — indicates the data movement state, and can have one of the following values:

    - Not Synchronizing
    - Synchronizing
    - Synchronized
    - Reverting pages from undo log
    - Initializing Undo log
- Synchronization Health — summarizes the synchronization status of all replicas in the availability group:
    - Healthy — if all replicas are fully synchronized.
    - Partially healthy — if only some of the replicas are fully synchronized.
    - Not healthy — if no replica is fully synchronized.
- Estimated Recovery Time (sec) — the estimated time that the secondary replica will require in order to catch up with the primary replica.
- Estimated Data Loss (sec) — the time difference of the last transaction log record between the primary and secondary replicas. Failure of the primary replica will result in the loss of all transaction log records within the time window.
- Is Failover Ready — indicates whether the secondary database is synchronized in the cluster with the corresponding primary database. Possible values are Yes or No.

Availability group map

The Always On configuration supports one primary replica and up to four secondary replicas for each availability group. The Availability Group Map section allows viewing the dependencies between the various replicas that form the availability group selected in the table. In addition to indicating the name of the OS cluster where the availability replicas reside, the map displays the connections between the primary replica and the secondary replicas, showing on the left the replica that resides on the currently monitored instance.

IMPORTANT: On each replica there is an indication of the highest severity level encountered on the replica.

A replica that is not currently monitored is indicated by the icon. Click this icon to launch the wizard used for configuring an instance for monitoring.

Clicking the monitored replica displays a pop-up with the following tabs:

- Group Info — identical to the Group Info tab on the pop-up that appears upon clicking the Explore column on the High Availability groups table. For details, see High Availability groups table on page 125.
- HA Summary — displays the selected replica's availability-related alarms, including the severity and the error message.
- Databases Health — a table displaying all alarms sent during the specified time range, including the severity and the error message.
- Listeners — if one or more availability group listeners are configured, displays the listeners' names and their IP status, either Online or Offline.

Clicking any other replica displays a pop-up with the following tabs:

- HA Summary — displays the selected replica's availability related alarms, including the severity and the error message.
- Databases Health — a table displaying all alarms sent during the specified time range, including the severity and the error message.

To view the dependencies between all replicas in all availability groups, click **View map for all availability groups**. To return from this map to the Always On panel, click **Back**.

Using the Logs Drilldown

The Logs drilldown provides access to SQL Server Error Log and SQL Server Agent Error Log.

To configure which error logs generated by the SQL Server database are to be displayed in the Logs drilldown, use the Error Log Scanning view in the Databases Administration dashboard. For details, see Defining Error Log Filtering on page 156.

Reviewing the SQL Server Error Logs

The SQL Server Error Logs panel displays errors that are defined in the match list and can be viewed by a paging resolution of 1 hour. This panel contains the following items:

- List of current and archived error logs

  Use this list to choose the error log whose details are to be displayed in the Error Log table below.

- Error Log table — displays the following parameters.

  Table 59. Error log table parameters:

Column	Description
Date/Time	The date and time when the specified error was logged.
Process Info	Information about the process to which the entry in the error log refers.
Message	A brief description of the error.

The SQL Server Error Logs table displays the contents of the error log selected in the list above the table. This table is a snapshot of the selected error log, and is therefore not updated automatically.

To define and edit the alert rules for which Foglight for SQL Server scans the SQL Server Error Log, go to the Error Log Scanning view in the Databases Administration dashboard. To access this view, click the **In-context actions** button at the upper right side of the screen and then select **Agent settings**. For details, see Defining Error Log Filtering on page 156.

Reviewing the SQL Agent Error Logs

The SQL Agent Error Logs panel displays errors that are defined in the match list and can be viewed by a paging resolution of 1 hour. It contains the following items:

- List of current and archived error logs

  Use this list to choose the error log whose details are to be displayed in the Error Log table below.

- Error Log table — displays the following parameters.

  Table 60. Error log table parameters.

Column	Description
Date/Time	The date and time when the specified error was logged.
Message	A brief description of the error. This description can also provide the reason for the error, thereby helping to resolve the issue; for example, *Unable to start mail session (reason: no mail profile defined).*

Reviewing Configuration Settings

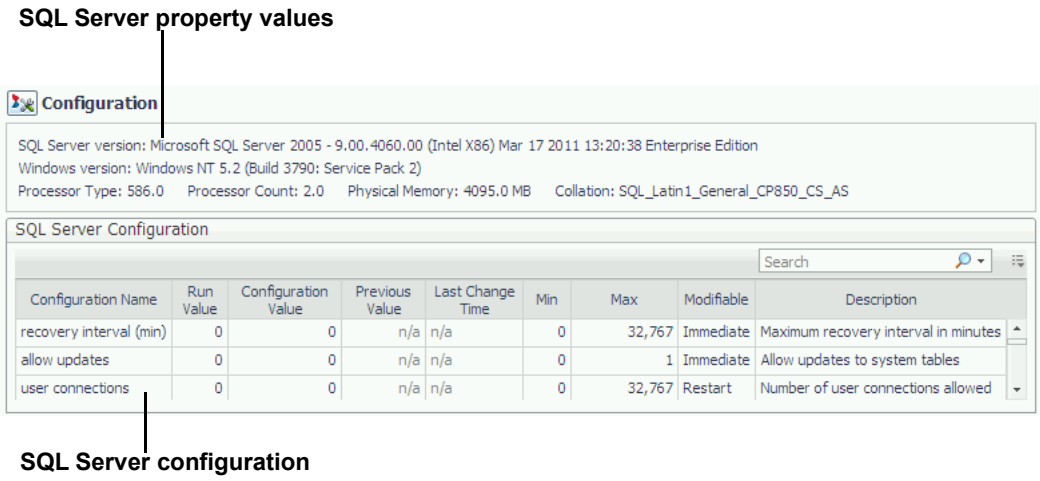
The Configuration drilldown displays the various SQL Server configuration options.

Reviewing SQL Server Configuration

The **SQL Server Configuration** panel allows carrying out the following tasks:

- Viewing the server property values — using the Server Property Values panel. See Viewing the Server Property Values on page 130.
- Viewing the SQL Server configuration options — using the Configuration table. See Viewing the SQL Server Configuration Options on page 131.

Figure 28. The SQL Server Configuration panel.



Viewing the Server Property Values

The **Server Property Values** panel contains information about the server properties of the monitored SQL Server instance.

Table 61. The Server Property Values parameters:

Parameter	Description
SQL Server Version	Version number of the currently monitored SQL Server instance.
Windows Version	The full version number of the Windows operating system installed on the computer that runs the SQL Server instance (including build and service pack).
Processor Type	The processor type (usually a vendor code).
Processor Count	The number of processors on the computer on which Windows is installed.
Physical Memory	The amount of actual RAM in the monitored host.
Collation	Default collation for databases on the SQL Server instance. Unless another collation is specified when creating a new database, this collation will apply to all of the instance's databases.

Viewing the SQL Server Configuration Options

The *SQL Server Configuration* table displays the configuration parameters for the currently monitored SQL Server, as listed in the table below. These parameters often affect the system's performance. Therefore, reviewing the values displayed in this table and modifying them, if needed, may successfully resolve performance issues.

To create a custom filter for this table, use the options accessible by clicking the button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 62. The SQL Server Configuration table parameters:

Parameter	Description
Configuration Name	The name of the specified configuration option.
Run Value	The value currently used by the configuration option.
Configuration Value	The value to which the configuration option is set. Depending on the Modifiable setting, changing the Config value can be implemented either immediately or only after restarting SQL Server.
Previous Value	The configuration option's value before the last change took place.
Last Change Time	The time the most recent change took place.
Min	The minimum permitted value for the configuration option.
Max	The maximum permitted value for the configuration option.
Modifiable	Indicates when changes to the option take effect. The following options are possible: • *Immediate*: changes to these options take effect immediately. • *Restart*: changes to these options take effect after SQL Server is stopped and restarted. • *Never*: these options cannot be modified.
Description	Description of the specified configuration option.

Viewing User-defined Performance Counters and Collections

The User-defined drilldown displays Performance Counters and Collections, plotted over the specified time range.

NOTE: This feature is only supported for monitored SQL Servers running on Windows platform.

The sections of this panel allow carrying out the tasks detailed in the following topics:

- Viewing User-Defined Performance Counters
- Viewing User-Defined Collections

Viewing User-Defined Performance Counters

The Performance Counters panel allows viewing the customized performance counters, which were created using the Performance Counters view in the Databases Administration dashboard, plotted over the specified time range.

Upon entering this panel, each user-defined performance counter is displayed on a separate row, with a counter that displays its current count.

To view a user-defined performance counter, click the arrow on the right side of the counter.

The Performance Counter pane appears, containing the following panels:

- *Current* — displaying the user-defined performance counter's current value of the selected counter unit (for example, second).
- *History* — displaying the performance counter's significant values in the specified time range:
  - Max — the counter's maximal value of the selected counter unit.
  - Avg — the collection's average value of the selected counter unit.
  - Min — the collection's minimal value of the selected counter unit.
- *Chart* — provides a visual representation of the performance counter's activity during the specified time range.

NOTE: The User Metrics drilldown is used only for displaying the user-defined performance counters; any creation or management operation of these counters is carried out using the Performance Counters view in the Databases Administration dashboard. For details, see Configuring Performance Counters on page 157.

Viewing User-Defined Collections

The Collections panel allows viewing the customized collections, which were created by means of the User-defined Collections view in the Databases Administration dashboard, either during the last sample (**Last Snapshot**) or plotted over the specified time range (**Selected Period**).

This panel comprises the following sections:

- The *Collections* column, on the left of the panel — displays the names of all existing user-defined collections.
- SQL text grid — displays the text of the user-defined SQL query.

  By default, this grid displays the query's short text. To display the query's full text, click **View full text** (a toggle).
- The collection details table — displays all fields contained in the query.

  Each of the table's columns shows the field's display name, while each row represents the records taken at each sample.

NOTE: The User-defined > Collections panel is used only for displaying the user-defined collections; any creation or management operation of these collections is carried out by means of the User-defined Collections view in the Databases Administration dashboard. For details, see Configuring User-defined Collections on page 162.

3

Monitoring Business Intelligence Services

Foglight for SQL Server Business Intelligence (BI) offers deep analysis monitoring solutions for SQL Server BI services: Reporting services, Integration services and Analysis services.

The Integration and Reporting services relay on monitoring the instances hosting the Reporting\Integration databases before installing the integration or reporting services agents.

The Analysis services agent requires using SQL PI capabilities and repository which will be installed during the initial Analysis agent installation, refer to the Deployment guide for the required resources.

This section includes these topics:

- Monitoring Integration Services on page 133
- Monitoring Reporting Services on page 136
- Monitoring Analysis Services on page 137
- BI Administration Settings on page 140

Permissions

- The same user monitoring the SQL Server database engine must be used to monitor the Integration and Reporting Services.
- The login ID used to monitor the Integration Service must be a user on the SSISDB database. This user ID is created while applying the “Grant permissions” script.
- The ID used to monitor the Integration Services on the database needs to have:
  - the `ssis_admin` role in order to gather all needed information for its collections.
  - the `db_datareader` role on the SSISDB database.
- Monitoring Analysis Services requires system administrator permissions on the Analysis Services instance.
- No additional permissions are required to monitor the Reporting Services.

Monitoring Integration Services

Configuring Integration Services Monitoring

Since Foglight for SQL server already monitors the SQL server instance you need only supply the server credentials.

IMPORTANT: A new agent could only be created when the SQL Server instance is already monitored.

To configure IS monitoring:

1 Click **Monitor**.

2 Select the instance from the list of monitored instances.

3 Select whether you want to generate the name automatically.

The default name of an Integration Service agent is `<Instance name>_<Repository name>`. The repository name for SSIS is always SSISDB.

4 To enable monitoring the Integration Service, provide the OS relevant credentials under "Integration Service Account":

a The name of the *underlined* host on which the service run.

In the case of a clustered SQL server, identify only the primary server. When a failover occurs the event might fail running packages and is treated as an availability alarm.

b The username and password allowed to view the service activity and performance counters.

Stored credentials could be used instead of supplying username and password manually.

5 Click **Monitor**.

Overview dashboard

The Foglight for Integration Server Overview dashboard provides a quick view of the current state of the instance.

The top of the dashboard includes general information on the monitored instance:

- **Schema build** - the exact build of the schema on the SSIS catalog.
- **Schema Version** - the exact version of the schema version.
- **Retention window** - number of days logged operations data is kept in the catalog database before it is deleted.
- **Logging level** - the default logging level used for the running packages.
- **Encryption algorithm level** - the security encryption algorithm used for storing the packages XML "scripts".

  NOTE: Foglight is unable to show the content of packages stored on the catalog since they are encrypted.

The dashboard tiles include a title that states the name of the monitored issue and an aggregation of the alarms relevant to that issue. Tiles are organized by priority:

- Availability - information and alarms about the availability of the instance. Also includes the monitoring state for that instance.
- HA/DR - High Availability (HA) & Disaster Recovery (DR) state of the instance. At this stage no HA issues are monitored. DR refers to the backup of the database used for the SSIS catalog.
- Storage - the space utilization level for the disk with the least free space on the server and the utilization level of the catalog database itself.
- Infrastructure - general information collected from the server and the service.
  - CPU - the Integration Service CPU level of utilization compared to the general utilization of the Integration Service.
  - Memory - the amount of memory used by the Integration Service compared to other processes and the total memory on the server.
  - Disk I/O - Displays the latency of the slowest disk on the monitored server. A High I/O value might lead to low IO rate for the running ETL processes (packages).
  - Network Utilization - the amount of network pressure experienced by the server.

- Operational - summary of the packages execution state over time. The panel displays:
  - The number of failed executions.
  - The number of successful executions.
  - Executions that ended in a state other than success or failure.

  Additional information is available on the Packages dashboard.
- Performance - The tile summarizes general information regarding the measurable performance of the packages. For example, total read and written bytes during a time range.

  The tile also displays the amount of total packages run during the time period examined and the amount of packages currently running. This number includes packages in running, pending or stopping status.

The alarms panel on the right side of the dashboard displays the list of alarms relevant to the monitored Catalog - Service pair. The errors can be organized by Time or Severity.

Packages Dashboard

The Packages dashboard displays the list of packages and their activity metrics over time. The data is collected directly from the catalog logs.

The dashboard provides information that allows you to:

- Detect failures during a specific time range.
- Detect executions that run for longer than normal during the time range selected.
- View the inactive packages during the time range selected.
- View the list of packages stored on msdb *.

NOTE: Packages on msdb are not analyzed yet and no log is read for them. Packages on the file system or package store are not monitored in this Foglight version. To analyze the performance of msdb packages, use the Jobs dashboard or the PI for the relevant job step if the step executes SQL statements.

The dashboard is composed of four components:

- The left pane, used to display information about package, executions and steps run during the time frame selected. This pane shows the last duration (for packages) or the actual duration (for executions or steps) in the given time range. It is also used to discover if there are excessive durations or failures during the time range.
- The trend graph in the right pane, used to discover executions trends over time, compare execution time to the normal behavior of the package executions and to view final status at a glance.
- General description of the package, execution, or step also appear in the right pane.
- Messages relevant to the package, execution, or step appear on a designated tab on the right pane.

  This tab can also be used to search the message log for information.

Terminology

Standard score - is used to compare the duration of a specific execution in compare to the normal execution.

Standard score is calculated as the current execution minus the average while the result is divided by the standard deviation. This score gives the result while considering the tendency of the execution duration fluctuation.

Step - is used to describe tasks, data flows and other package components. Steps are used to discover which part of the package has failed or exceeded their normal duration.

Monitoring Reporting Services

Configuring Reporting Services Monitoring

IMPORTANT: A new agent can only be created when the SQL Server instance is already monitored.

To configure RS monitoring:

1. Click **Monitor**.
2. Select the instance from the list of monitored instances.
3. Select whether you want to generate the name automatically. The default name of a Reporting Service agent is `<Instance name>_<Report_DB>`.
4. To enable monitoring the Reporting Service, provide:
    a. the OS relevant credentials under Reporting Service Account, which is the name of the underlined host on which the service runs.
    b. The username and password allowed to view the service activity and performance counters. Stored credentials can be used instead of entering the username and password manually.
5. Click **Monitor**.

Monitoring Reports using the Reporting Services Panel

The metadata of the reporting services is stored in the SQL Server instance. Foglight for SQL Server offers the capability of analyzing the metadata and produces detailed reports of the user's activity.

This panel contains the following components:

- Reporting Services Databases — allows selecting which metadata database is to be used.
- Reporting Services Configuration — allows reviewing the configuration of the metadata databases selected in the Reporting Services Databases section. The settings displayed in this section are read-only and cannot be modified.
- Distribution of Reported Runs — a pie chart displaying the distribution of the report runs between successful and failed runs.
- Top Users — displays the users who generated the highest number of reports.
- The Reporting Services Execution Summary table — displays a row for each report, and includes the following columns:
    - Path — the path where the report resides.
    - Report Name — the name of the report.
    - Status — the report's status, that is: the result of the report's running; either Failed or Success.
    - Total Time Retrieving Data — the total time required for collecting the report's data.
    - Total Time Processing — the total time required for processing the data retrieved for the report.
    - Total Time Rendering — the total time required for rendering the data for UI display.
    - Total No. of Rows — the total number of rows displayed in the report.
    - No. of Executions — the total number of times the report was executed.

- Explore — allows reviewing in-depth data about the report.

Reporting Services Execution

The Reporting Services Executions pop-up displays a different row for each run, each row showing the following columns:

- User Name — the name of the user who ran the report
- Start Time — the time that the report run's started
- End Time — the time that the report run's ended
- Data Retrieval Time — the total time required for collecting the report's data
- Processing Time — the total time required for processing the data retrieved for the report
- Rendering Time — the total time required for rendering the data for UI display
- Status — the report's status, that is: the result of the report's running; either Failed or Success
- No. of Rows — the number of rows displayed in the report
- Format — the report's format
- Parameters — displays the parameters that were included during the report's run

Monitoring Analysis Services

Foglight for Analysis Service is bundled with the Performance Investigator tool. When adding an Analysis Service agent a new repository for Performance Investigator will be created as well.

Configuring Analysis Services Monitoring

To add a new agent:

1. Click **Monitor**.
2. Under **Connection details**, type the server name. If the server name includes the named instance separate the with a backslash: `<servername>\<instance>`.
3. Specify the login credentials or use an account already used by the agent manager.
4. Select "Automatically generate agent name" or write a unique name.
5. Enter the OS credential, if different than the instance login credentials.

   You can also use credentials from Foglight's stored credentials. Stored credentials can be used instead of providing the username and password

Overview dashboard

The Foglight for Analysis Server Overview dashboard provides a quick view of the current state of the instance.

The top of the dashboard includes general information on the monitored instance. Each tile is constructed of a title that state the name of the monitored issue and an aggregation of the alarms relevant to that issue.

The tiles are organized by priority:

- Availability - information and alarms about the availability of the instance. Also includes the monitoring state for that instance.

- HA/DR - High Availability (HA) & Disaster Recovery (DR) state of the instance. At this stage no HA issues are monitored. DR refers to the discovered backups on the commands that are captured by Foglight's Performance Investigator (PI).
- Storage - the total space available on the monitored server in comparison to the space used by Analysis Services databases (estimated).
- Infrastructure - general information collected from the server & the service.
    - CPU - Show the Analysis Service CPU level of utilization in compare to the general utilization of the instance.
    - Memory - Shows the Analysis Service memory usage in compare to other processes and the total memory on the server.
    - Disk I/O - Displays the latency of the slowest disk on the monitored server.
    - Network Utilization - Show the amount of network pressure experienced by the server.
- Operational - the processing state of the instance.
    - The number of unprocessed objects - show the number (up to 500) of objects that are in an unprocessed state.
- Performance - summarizes general information regarding the workload rates (time proportion for each discovered wait category, for example, Lock, CPU, IO, Memory). The tile also display general information regarding the throughput and the sessions' state.

The alarms panel to the right side of the dashboard displays the list of alarms relevant to the monitored instance. The errors can be organized by Time or Severity.

Unprocessed Objects Dashboard

This dashboard displays the list of objects that are marked as being in an unprocessed state and cannot be accessed. The list is being calculated on demand, which may cause some delay when opening the dashboard or refreshing it.

Current Activity Dashboard

The current activity dashboard displays the list of current sessions and their state. The dashboard can be used to examine the amount of sessions connected, the status of the sessions and amount of resources they used.

Monitoring Analysis Service Performance

The Performance Investigator (PI) dashboard provides the ability to investigate the activity and resource consumption of a selected instance. The Analysis Service PI provides a more in-depth analysis by adding a dimension view of the activity, blocking history and change tracking analysis.

For additional information, refer to:

- Performance Tree on page 139
- Viewing Historical Metrics on page 140
- Viewing Change Tracking on page 140

Resource Toolbar Options

The SSAS PI dashboard displays the following components:

- The Instance View — Provides the ability to investigate and analyze the resource consumption of the instance.
- The Resource toolbar — By selecting one of the resources, the resource consumptions charts and the overview section are updated to display only the relevant information regarding the activity of the selected resource.

The following resource are available:

- Workload — Allows you to review the overall workload by gathering all resources into one view.
- CPU — Allows you to review CPU usage.
- I/O — Allows you to review the I/O-related data, such as wait time for I/O, reads and writes.
- Memory — Allows you to review memory related waits and performance data, such as memory used, memory cache size and more.
- Lock — Allows you to review blocks and deadlocks related information and identify blocked activities.
- Remote — Allows you to review amount of time activities spent waiting for processing to finish on remote instances such as relational data sources

To review specific data metrics, select one of the specific metrics from the toolbar.

Performance Tree

The performance tree provides iterative (up to three levels) access to any of the key dimensions associated with

Analysis Services activity, based on the OLAP multidimensional model and an instance view of the instance activity. Domain nodes offer a hierarchical view of all types of SSAS activity characteristics. Selecting a dimension from the tree determines what subset of activity is displayed. Iterative drill-downs into domains of interest provides increasingly refined focus and diagnosis.

For example, to begin the investigation by first identifying the most active User, follow the steps described below:

1. Select the Users node, to display the most active users in the selected time range. That is, the users who consumed the highest amount of the selected resource.
2. Select the first user, to focus the entire window on the activity of that user.
3. Identify the most demanding Command that this specific user has executed, by expanding the user node and then selecting the Command dimension node. This displays the most active SQL statements executed by this user.
4. Select a specific Command to focus the entire window on the selected command's activity.
5. Select Client Machines under the selected Command, to view the computers which initiate the command.

In a similar manner, such iterative drilldowns can be carried out into any SSAS dimension of interest, to gain a complete understanding of the causes of its behavior.

The default Analysis Services dimensions are as follows:

- Commands — The executed MDX, DMX and XMLA commands.
- Databases — The database context in which the session carried out its operations. A session may switch to numerous databases within its lifetime.
- Applications — Name of a program that connects to SSAS and executes the Command.
- Users — Users running the client program.
- Client Machines — The machines on which the client executable (connected to SQL Server) is running.
- Command Types — Executed command type (for example, Batch, XMLA, SQL etc).
- Sessions — Presents the top sessions which consumed the highest active time during the selected time frame.

- Objects — Displays the top objects that were used by all activities running on the server,their association to other dimensions and the resources they used.

In the Commands dimension there is an option to view information regarding the selected command through the top Commands table top:

- View Full Text — View full text of the statement\batch selected.

Viewing Historical Metrics

The History section view is divided into two sections that are correlated to each other:

- Resource consumption charts — This section displays data in five different charts:
  - Workload chart — Displays the instance resource activity over the selected time frame by emphasizing the resources by colors.
  - Breakdown chart — Activity of the instance by second.
  - Resource Breakdown Pie chart — Displays the resource breakdown usage by % of the total instance activity.
- Overview section — Displays a graphical representation of the metrics highlighted in the Workload related Metrics table.
  - Workload related Metrics — A table that displays a variety of resource consumption metrics which can give an in-depth of the instance activity, each resource holds its default metrics.

Selecting each dimension in the performance tree together with a specific resource effects the data displayed for each Level. For example, by selecting the Lock resource, the Instance view dimension will present only locks related data, the Command dimension will present only the commands that were experiencing locks , Users will show only users that were experiencing locks and so on across all the dimensions and resources.

Viewing Change Tracking

The change tracking tool is an integrated monitoring mechanism. It periodically tracks changes in environments and activity that can potentially influence system performance. Use it to view correlation between occurrences of changes and Analysis Services activity and behavior patterns.

Use the Categories filter mechanism to refine the set of displayed change tracking occurrences. These categories are displayed on the right hand side of the pane and include:

- MetaData — Report on changes to the Analysis Services server configuration and to the underlying database's schema.
- System Configuration — Hardware and operating system configuration (for example, disks and network interfaces (NIC), amount of RAM, CPU count, device installations and swap space allocation).
- User defined — Used for manually documenting user changes that may affect performance. Recording this activity can assist in determining whether a change in performance can be directly associated with this event

BI Administration Settings

To change the administration settings for multiple agents of the same type:

1. Select multiple agents of the same type: Integration, Reporting or Analysis services.
2. On the databases table toolbar, select **Settings>Administration**.
3. Select a category of settings on the left: Alarms or Connection Details.

A view containing related settings opens. The definitions under the Administration dashboard apply to all of the agents selected before opening the Administration dashboard.

4 Save changes made to the settings before selecting another category of settings.

NOTE: To view the selected agents click the Selected Agents button at the upper right corner of the screen. If the settings vary between the selected agents the fields that contain non-identical values are displayed as empty and marked with an Inconsistent Values icon.

Defining Connection Details

Use the Connection Details category to redefine connection settings, which apply to one or all of the selected services hosts.

Alarms

The Alarms view enables you to modify global settings and agent-specific settings for alarms.

To open the Alarms view:

1 Select the agents you wish to modify its alarms settings.

2 Do one of the following steps:

- Select the **Settings** button and open the Administration dashboard, then click **Alarms**.

  OR

- Select the Configure Alarm' button.

3 From the Alarms view, you can complete the following tasks:

- Modifying Alarm Templates on page 148
- Configuring Email Notifications on page 150
- Cloning Agent Settings on page 154

4

Administering Foglight for SQL Server

Foglight for SQL Server monitors the SQL Server database activity by connecting to and querying the SQL Server database. The agents provided monitor the SQL Server database system. The dashboards included with the cartridge provide a visual representation of the status of the major components of the SQL Server agents. They allow you to determine any potential bottleneck in database performance.

Configuration Settings

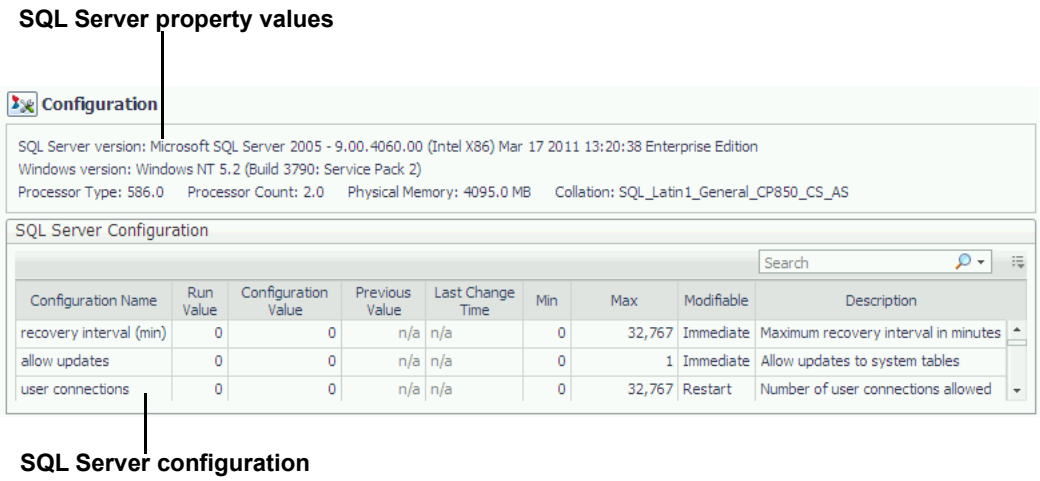
The Configuration drilldown displays the various SQL Server configuration options.

Reviewing SQL Server Configuration

The **SQL Server Configuration** panel allows carrying out the following tasks:

- Viewing the server property values — using the Server Property Values panel. See Viewing the Server Property Values on page 142.
- Viewing the SQL Server configuration options — using the Configuration table. See Viewing the SQL Server Configuration Options on page 143.

Figure 29. The SQL Server Configuration panel.



Viewing the Server Property Values

The **Server Property Values** panel contains information about the server properties of the monitored SQL Server instance.

Table 63. The Server Property Values parameters:

Parameter	Description
SQL Server Version	Version number of the currently monitored SQL Server instance.
Windows Version	The full version number of the Windows operating system installed on the computer that runs the SQL Server instance (including build and service pack).
Processor Type	The processor type (usually a vendor code).
Processor Count	The number of processors on the computer on which Windows is installed.
Physical Memory	The amount of actual RAM in the monitored host.
Collation	Default collation for databases on the SQL Server instance. Unless another collation is specified when creating a new database, this collation will apply to all of the instance's databases.

Viewing the SQL Server Configuration Options

The *SQL Server Configuration* table displays the configuration parameters for the currently monitored SQL Server, as listed in the table below. These parameters often affect the system's performance. Therefore, reviewing the values displayed in this table and modifying them, if needed, may successfully resolve performance issues.

To create a custom filter for this table, use the options accessible by clicking the button at the table's upper right side. For details, see Components Shared by all Foglight for SQL Server Screens on page 28.

Table 64. The SQL Server Configuration table parameters:

Parameter	Description
Configuration Name	The name of the specified configuration option.
Run Value	The value currently used by the configuration option.
Configuration Value	The value to which the configuration option is set. Depending on the Modifiable setting, changing the Config value can be implemented either immediately or only after restarting SQL Server.
Previous Value	The configuration option's value before the last change took place.
Last Change Time	The time the most recent change took place.
Min	The minimum permitted value for the configuration option.
Max	The maximum permitted value for the configuration option.
Modifiable	Indicates when changes to the option take effect. The following options are possible: • *Immediate*: changes to these options take effect immediately. • *Restart*: changes to these options take effect after SQL Server is stopped and restarted. • *Never*: these options cannot be modified.
Description	Description of the specified configuration option.

Managing Foglight for SQL Server Agent Settings

You use the Databases Administration dashboard to set options for collecting, storing, and displaying data about monitored SQL Server instances.

Opening the Databases Administration Dashboard

You can edit agent settings for one or more SQL Server instances on the Databases > Administration dashboard.

NOTE: If you attempt to select instances of more than one type of database, such as an SQL Server database and an Oracle database, an error message is displayed.

To open the Databases Administration dashboard:

1. In the navigation panel, under **Homes**, click **Databases > SQL Server**.
2. Select the check boxes beside one or more SQL Server instances.
3. Click **Settings** and then click **Administration**.

   The Administration dashboard opens, containing settings for all the selected agents. Settings are broken down into categories, which are organized under a SQL Server tree.

Reviewing the Administration Settings

The Databases Administration dashboard allows settings options for collecting, storing, and displaying data, which apply to all the currently selected agents. Click a category of settings on the left (for example: Connection Details) to open a view containing related settings on the right.

The metrics defined under the Databases Administration dashboard apply to all of the agents that were selected before opening the Databases Administration dashboard. As a result, the same unit of measure and aggregation value for display are enforced for all currently selected agents.

The SQL Performance Investigator category allows you enable or disable SQL PI for the agents selected. In addition, you can also start or stop change tracking for the agents.

To view the full list of selected agents, click the **Selected Agents** button at the upper right corner of the screen. To change the list of agents to which the metrics will apply, exit the Databases Administration dashboard, select the requested agents and re-open the view.

If the settings vary between the selected agents (for example: one agent uses the measurement unit kilobyte, while another uses megabyte), the fields that contain non-identical values are displayed as empty and marked with an Inconsistent Values ( ) icon.

Changes made to settings should be saved before selecting another category of settings.

To save changes made in an Administration dashboard view:

1. In the Database Administration dashboard, select a category from the menu.
2. Make changes to settings as necessary.
3. Click **Save changes** at the bottom of the view.

   If you attempt to exit the view without saving changes, a Warning dialog box prompts you to confirm your action.

Defining Connection Details

Use the Connection Details category to define global connection settings, which apply to all instances and hosts selected in the view. You can configure SQL Performance Investigator connectivity, enable user-defined collections, and set VMware connection details.

Defining the Connection Settings for the Monitored Instances

The Connection Details view contains a table that displays all the agents that were selected before entering the Databases Administration dashboard.

To define the connection settings for the requested agents:

1. Select the check boxes to the left of the agents for which uniform credentials are to be set.
2. Click **Settings,** and then click **Administration**.

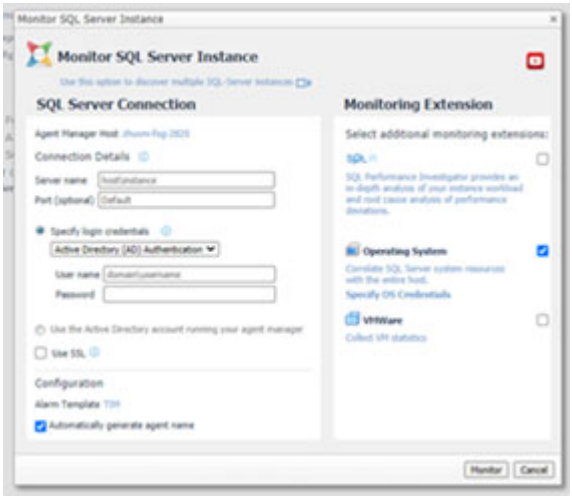
   The Administration dashboard opens, containing settings for all the selected agents.
3. Click **Connection Details**, and then click **Set credentials**.

   The *Set* dialog box used for editing the credentials of the selected instance appears.

IMPORTANT: If multiple instances were selected before clicking Set credentials, this dialog box is empty (does not display the instances' names, ports, and connection details). The default options in such a scenario are Windows authentication (for SQL Server connection details) and the use of existing host connection details (for OS monitoring).

4. On the *Edit Credentials* pane, provide port connection details. This field can be left empty, unless the TCP/IP connection port is not the default port: 1443.
5. Specify the SQL Server login credentials using either of the following authentication methods:
   - Active Directory (AD) Authentication — log in using the Active Directory account running on your agent manager or typing a new AD account. The user name should be typed in the following format: domain\user name.
   - SQL Server Authentication — log in using a SQL Server account
6. Use SSL: Select whether an SSL connection should be used.

Figure 30. The *Monitor SQL Server Instance* Panel



7 In the Monitoring Extensions section you can select and configure which monitoring extension to add to the monitored instance:

- Operating System — Correlate the SQL Server OS resources with the entire host.

NOTE: Note: Monitoring OS can made also by using stored credentials - Select this link to open the Stored Credentials dialog box, which allows reviewing the login credentials and authentication methods used for logging in to Foglight. Foglight stores encrypted credentials in lockboxes, which may be password-protected for added security. The Credentials dialog box, DB-Agent Lockbox. If credentials have already been entered in another lockbox, use the Lockbox list to select from that lockbox.

- VMWare — Collect VM statistics.

8 Click **Set.**

9 On the Connection Details pane, click **Test Connection**.

NOTE: If the monitoring verification fails click the message that is displayed on the Status column and resolve the issue according to the instructions that appear in the dialog box. For example, insufficient privileges, incorrect credentials or an Agent Manager that reached its full monitoring capacity.

Upon successful completion of the process, the status shows as Verified.

10 The option of enabling and editing credentials for user-defined collections is available also when multiple instances are selected. This option is carried out as follows:

a Click **Set UDC Credentials**.

The dialog box Edit Credentials for User-defined Collections appears.

b Select the check box Enable user-defined collections.

c Select whether to perform the collection:

- Using the current agent credentials
- Manually specifying login credentials.
- Using Windows authentication and the Windows account that is running Foglight.

d If the option of manually specifying login credentials was chosen, select the requested authentication method from one of these options:

- Windows authentication — type a user name and a password in the domain\username format (for example, COLUMBIA\JSmith).
- SQL Server authentication

IMPORTANT: If the Foglight Agent Manager that runs the instance resides on a UNIX host, the option of using Windows authentication and the Windows account that is running Foglight is unavailable.

e Click **Set** to return to the dialog box used for editing the instance's credentials.

IMPORTANT: To add user-defined collections, go to the User-defined Collections view in the Databases Administration dashboard. For details, see Configuring User-defined Collections on page 162.

11 If you have SQL Server instances that run on virtual hosts, such instances require setting a dedicated connection profile, in order to connect to the requested VMware server.

Establishing such a connection is necessary in order to retrieve the Virtualization overhead data, that is, the percentage of CPU that is unavailable to this virtual machine because it is being utilized either by other virtual machines or by VMware itself. The Virtualization Overhead indicator is displayed in both the real-time and history summary pages.

To edit the VMware Connection Profile:

a Select the requested agents. To select all agents, select the check box on the table's title column.

b Click **Edit**.

The Edit VMWare Credentials dialog box appears.

c Select the check box *Enable collecting VMWare CPU allocation data*.

d Enter the details required for monitoring the CPU distribution data, that is, host, port, VMware user, and VMware password.

e Click **OK**.

If multiple agents were selected, the settings will apply to all agents.

12 Ensure that all requested data has been entered. If so, click **Test Connection**.

A progress bar appears.

At the end of this process, any connectivity issues are indicated by the *Status* column of the agents table. This column displays either the status *Verified* for the instances that connected successfully to the database, or a status that indicates failure of the connectivity verification process, and the reason for the failure (for example, *Login failed for user "X"*).

If the connectivity issue results from faulty login credentials, modify the credentials and carry out again the connection verification process. If the database to which the instance tries to connect is not running, clear the box near the database's name.

If some of the instances whose verification failed display a status of either *Insufficient privileges, Click to Grant* or *Wrong sysdba Credentials*, such instances should be granted privileges, by clicking the status and using one of the following methods:

- Manually, using a script (by clicking **View script,** copying the text and using it to grant privileges
- By clicking the button **Grant privileges**.

The Grant Privileges dialog box appears.

If Insufficient Privileges are indicated, this dialog box allows specifying a SYSAdmin (System Administrator) user with sufficient privileges.

Enter a SYSAdmin user and password, and then click **Grant Privileges**.

13 Click **Save Changes**.

The *Applying Modified Settings* progress bar appears.

Upon successful completion of this process, the *Status* column of the instance table displays the status *Changes applied*. For instances that failed verification, the status column indicates that changes cannot be saved if the validation did not complete successfully.

Customizing Alarms for Foglight for SQL Server Rules

NOTE: Foglight 6.0 introduces Alarm Templates, which provide a simplified method for customizing alarm rules and applying them to agents. As part of this change, the Sensitivity Level feature has been deprecated.

Many Foglight for SQL Server rules trigger alarms. To improve your monitoring experience, you can use alarm templates to customize when alarms are triggered and whether they are reported. You can also set up email notifications.

This section covers the following topics:

- Introducing the Alarms View
- Viewing Alarm Sensitivity Levels for Unassigned Agents
- Modifying Alarm Templates
- Configuring Email Notifications

- Cloning Agent Settings
- Reviewing Rule Definitions

Introducing the Alarms View

The Alarms view enables you to modify global settings and agent-specific settings for alarms.

To open the Alarms view:

1 Open the Administration dashboard as described in Opening the Databases Administration Dashboard on page 144.

2 Click **Alarms**.

The list of agents that you selected on the Databases dashboard is shown in the upper right corner of the view.

3 From the Alarms view, you can complete the following tasks:

- Viewing Alarm Sensitivity Levels for Unassigned Agents
- Modifying Alarm Templates
- Configuring Email Notifications
- Cloning Agent Settings

Viewing Alarm Sensitivity Levels for Unassigned Agents

In previous versions, Foglight for SQL Server offered four sensitivity levels that controlled which alarms were reported:

- **Essential** — Store and display only critical or fatal alarms.
- **Normal** — Store and display most alarms — essential and best practices; only critical and fatal statistical alarms.
- **Tuning** — Store and display all SQL Server alarms sent to Foglight.
- **Performance** — Store and display only availability and SQL PI related alarms.

This feature has been deprecated in Foglight 6.0. These sensitivity levels can no longer be changed, but for agents that have not been assigned to an alarm template, you may view a report detailing the sensitivity levels previously configured.

To view the sensitivity level used by an agent:

NOTE: You may only view sensitivity levels for an agent that has not been assigned an alarm template.

1 In the Alarms view, click View sensitivity level as PDF in the upper right corner.

A **Rules Sensitivity Report** will be displayed.

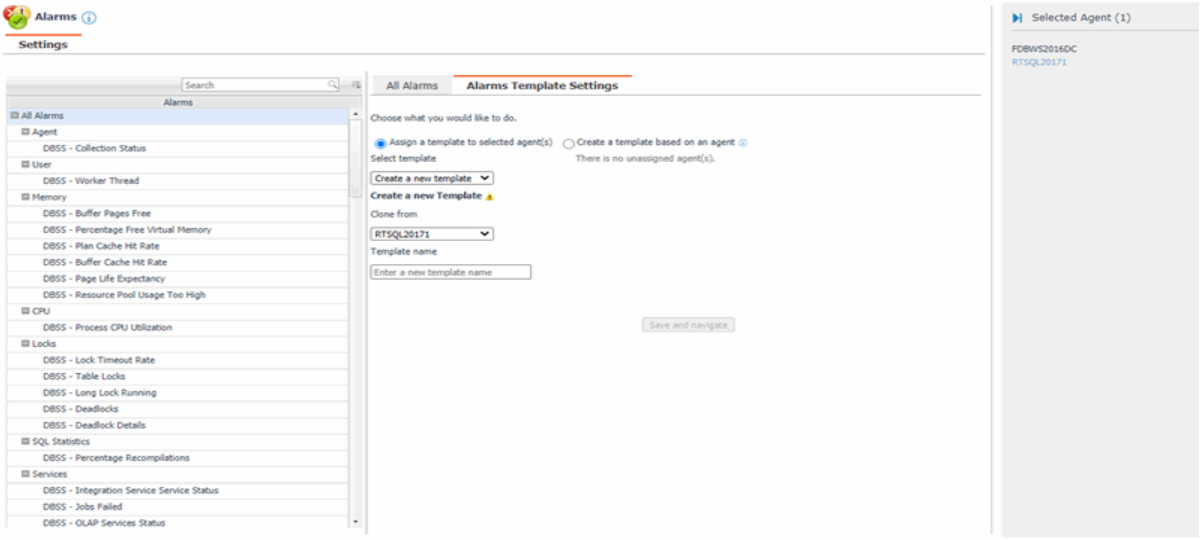
Modifying Alarm Templates

Foglight 6.0 uses alarm templates to gather alarm rules into a domain-specific template that is easily modified and applied to targets. You can customize how the alarms generated by the default Foglight for SQL Server rules are triggered and displayed in by assigning alarm templates via the **Alarm Template Settings** tab. You can apply an existing template, create a new template using an existing template as reference, or create a template based on an agent. All changes to alarm templates apply to the selected agents.

Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.0 User Guide* for more information.

IMPORTANT: Avoid editing Foglight for SQL Server rules in the **Administration > Rules & Notifications > Rule Management** dashboard. Default rules may be modified during regular software updates and your edits will be lost. Always use the Alarm Templates dashboard.

Figure 31. Use the Alarms view to assign alarm templates.



The Alarms list controls the contents displayed to the right and the tasks that are available.

- **All Alarms –** Displays all rules with configured alarms and indicates whether alarms are enabled. In this view, you can enable or disable alarms for all the rules at once. You can also set email notifications and define mail server settings.W hen viewing all alarms, the Alarm Template Settings tab is displayed, enabling configuration of alarm templates.
- **Category of rules –** Displays a set of related rules with configured alarms. In this view, you can set email notifications for the category of rules.
- **Rule name –** Displays the **Email Notification Status** for the selected rule. If the rule has multiple severity levels, displays the notification configured for each severity level. In this view, you can enable or disable email notifications for the alarm and edit alarm messages.

You can complete the following tasks:

- Assign an Alarm Template to selected agents
- Modifying alarms and threshold values

Your changes are saved separately and applied over the default rules. This protects you from software upgrades that may change the underlying default rules.

Assign an Alarm Template to selected agents

You can override the alarm rules for the selected agents by assigning an alarm template. You can use the template to enable or disable alarms for all rules or an individual rule, or to change the threshold values of an alarm rule.

To see descriptions of the rules, follow the steps described in Reviewing Rule Definitions on page 155.

To assign an existing alarm template:

1 In the Alarms view, click the **Alarm Template Settings** tab.

2 Select **Assign a template to selected agent(s)**,

3 Do one of the following:

- Select an alarm template from the drop-down.

- Click **Save and Navigate.**

OR

- Select **Create a new template** from the drop-down. This will allow you to create a new template based of an existing template.
- In the **Clone from** field, select the template to copy.
- Enter a name for the new template.
- Click **Save and Navigate.**

4 The template will be displayed in the Alarm Template dashboard. Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.0 User Guide* for more information on editing alarm templates.

Modifying alarms and threshold values

You can and should modify the thresholds associated with alarms to better suit your environment. If you find that alarms are firing for conditions that you consider to be acceptable, you can change the threshold values that trigger the alarm. You can also enable or disable severity levels to better suit your environment.

When a rule has severity levels, a Threshold section appears in the Alarm Settings tab showing the severity levels and bounds by agent. For an example, see the *DBSS - Worker Thread* rule. The threshold values corresponds to the lower bounds shown in this table. Many rules, such as Baseline rules, do not have severity levels and thresholds.

When editing thresholds, ensure that the new values make sense in context with the other threshold values. For most metrics, threshold values are set so that Warning < Critical < Fatal. However, in metrics where normal performance has a higher value, such as *DBSS - Buffer Cache Hit Rate*, the threshold values are reversed: Warning > Critical > Fatal.

To change alarm status and threshold values:

IMPORTANT: The procedure below is a summary. Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.0 User Guide* for more information on editing alarm templates for more information on working with alarm templates.

1 In the **Navigation** panel, click **Alarm Templates**.

2 If you have previously configured an alarm template, select that template.

Otherwise, click the Factory Template to view the default rules. Duplicate the factory template to make an editable copy, selecting the appropriate domains.

3 Click the appropriate domain tab (SQL Server, SQL Analysis Services, etc).

4 Scroll or search to find the alarm rule you want to edit. Click the rule to select it.

5 Edit the rules using the procedure described in **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.0 User Guide*

Configuring Email Notifications

We recommend that you set email notifications for the alarms you are most interested in tracking closely. For example, you may want to be notified by email of any Critical or Fatal situation. Or you may want to be informed whenever a key metric, such as CPU usage, is no longer operating within acceptable boundaries.

You can set up email notifications that are generated when an alarm fires and/or on a defined schedule, as described in the following topics:

- Configuring an email server
- Defining Default Email settings
- Defining email notifications, recipients, and messages

- Defining variables to contain email recipients
- Defining scheduled email notifications

Configuring an email server

You need to define the global mail server variables (connection details) to be used for sending email notifications. The setting of the email should be configured in **Foglight Administration > Email configuration**.

Defining Default Email settings

You can define a default email address to be used by every new agent created in the future, by selecting the Default email button when configuring email notification.

The Email addresses entered are applied to all monitored agents not only for the agents that were selected to enter the Alarm administration.

Enabling or disabling email notifications

You can enable or disable email notifications for all alarms, a category of alarms, or a selected rule. Email notifications are sent only if all the following conditions are met:

- The alarm email notification setting is enabled for the affected rule.
- The alarm is triggered by changes in the monitored environment.
- Alarm notification is enabled at the triggered severity level. See Defining email notifications, recipients, and messages.

To enable or disable email notifications:

1 In the Alarms view, click the **Settings** tab.

2 Decide on the scope for the change: all alarms, a category of rules, or a selected rule.

3 Complete the steps for the selected scope:

Table 65. Enable or disable email notification settings.

Scope	Procedure
All alarms	Click **All Alarms**. In the Alarms Settings tab, click the **Define Email Settings** button. Select either **Enabled** or **Disabled** from the Alarms notification status list. Click **Set**.
Category of rules	Click a category. Click the **Define Email Settings** button. Select either **Enabled** or **Disabled** from the Alarms notification status list. Click **Set**.
Selected rule	Click a rule. In the Alarms Settings tab, click the **Define Email Settings** tab. Click the link that displays the alarm notification status. Select **Enabled** or **Disabled** and click **Set**.

4 Click **Save changes**.

Defining email notifications, recipients, and messages

You control who receives email messages, the subject line, and some text in the body of the email. The body of the email always contains information about the alarm. This information is not editable. You can also control whether an email is sent based on severity levels. You can set different distribution lists for different rules and different severity levels, or set the same notification policy for all rules.

To configure email notifications:

1 In the Alarms view, click the **Settings** tab.

2 Decide on the scope for the change: all alarms, a category of rules, or a selected rule.

3 Complete the steps for the selected scope:

Table 66. Configure email notification settings:

Scope	Procedure
All alarms	Click **All Alarms**. In the Alarms Settings tab, click the **Define email settings** button. Continue to Step 4.
Category of rules	Click a category. Click the **Define email settings** button. Continue to Step 4.
Selected rule	Click a rule. Click the **Define email settings** tab. • To change the severity level that warrants an email notification, click the link that displays the severities. Select the desired level of severity and click **Set**. • To configure email recipients and the message, select the tab for a severity level, and click **Edit**. Skip to Step 5.

4 If you selected **All Alarms** or a category, in the Email Notification Settings dialog box, do one of the following:

- To change the severity levels that warrant an email notification, from the **Messages will be enabled for severities** box, select the desired levels of severity.
- To configure the same email recipients and message for all severity levels, click **Configure mail recipients for all Severities** and then click **All severities**.
- To configure different email recipients and messages for each of the severity levels, click **Configure mail recipients for the following options** and then click a severity level.

5 In the Message Settings dialog box, configure the email recipients and message.

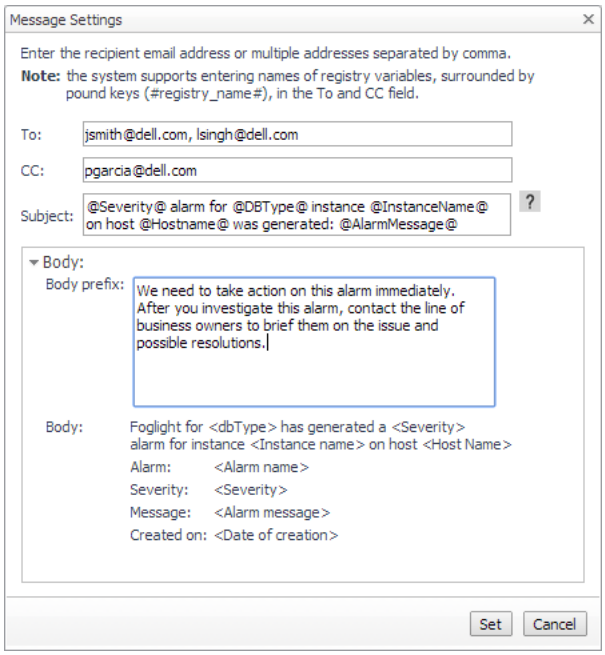
- **To** — Type the addresses of the people who need to take action when this alarm triggers.
- **CC** — Type the addresses of the people who want to be notified when the alarm triggers.

NOTE: If a mail server is not found, you are prompted to configure a mail server. For instructions, see Configuring an email server on page 151.

You can use registry variables in place of email addresses. Type the variable name between two hash (#) symbols, for example: #EmailTeamName#. For more information, see Defining variables to contain email recipients on page 153.

- **Subject** — Optional. Edit the text of the subject line to better suit your environment. Avoid editing the variables, which are identified with the @ symbol.
- **Body Prefix** — **Optional**. Add text that should appear above the alarm information in the body of the email.

Figure 32. Message Settings



6 Click **Set** to save the message configuration and close the dialog box.

7 If the Edit Notification Settings dialog box is open, click **Set**.

8 Click **Save changes**.

Defining variables to contain email recipients

You can create registry variables that contain one or more email addresses and (optionally) their scheduled notifications, and use these registry variables when defining email notifications. This procedure describes how to create a registry value. For schedules, see Defining scheduled email notifications.

To create a registry variable:

1 On the navigation panel, under Dashboards, click **Administration > Rules & Notifications > Manage Registry Variables**.

2 Click **Add**.

The New Registry Variable Wizard opens.

3 Select the registry variable type **String**, and click **Next**.

4 In the Name field, enter a name, for example: **EmailTeamName**

Optional — Add a description.

5 Click **Next**.

6 Select **Static Value**.

7 In the Enter desired value box, enter one or more email addresses (separated by commas).

NOTE: Email groups are not permitted.

8 Click **Finish**.

The Edit Registry Variable dashboard displays the newly created registry variable.

To use a registry variable in email notifications, type the variable name between two hash (#) symbols, for example: **#EmailTeamName#**. For more information, see Defining email notifications, recipients, and messages on page 151.

Defining scheduled email notifications

If someone wants to receive an email about an alarm on a regular basis, such as once a day, you use a registry variable schedule to set up the notification.

To schedule the sending of email notifications for a registry variable:

1. If you are continuing from Defining variables to contain email recipients, the registry variable is already open for editing in the Edit Registry Variable dashboard.

TIP: To edit a different variable, navigate to the **Administration > Rules & Notifications > Manage Registry Variables** dashboard, click the variable name, and select **View and Edit Details**.

2. In the Performance Calendars List table, click **Add**.

   The Performance Calendar Wizard opens.

3. Select a schedule, for example: **End of Day**
4. Click **Next**.
5. Select **Static Value**.
6. In the Enter desired value box, enter one or more email addresses (separated by commas) that should received email notifications based on the schedule.

TIP: The addresses may the same as or different from those assigned to the registry variable.

7. Click **Finish**.

   The Edit Registry Variable dashboard displays the newly created schedule. If desired, repeat to add other schedules.

Cloning Agent Settings

You may want an agent to have the same settings as another agent. For example, if you add new agents, you may want them to use the same settings as an existing agent. In this case, you can clone the settings from one agent to other agents. This process does not link the agents; in the future if you update the source agent, you also need to update the target agents.

This procedure walks you through selecting the source agent from the Databases dashboard. However, you can also open the Administration dashboard with multiple agents selected. In this case, you select the source agent in Clone Alarm-related Settings to Other Agents dialog box.

To clone alarm-related settings:

1. On the Databases dashboard, select the check box for the agent with the settings you want to clone.
2. Click **Settings** and then **Administration**.
3. In the Administration dashboard, click **Alarms**.
4. Click **Set configuration on selected agents**.

   The Clone Alarm Settings cross Agents dialog box opens.

5. In the Select the source agent drop-down list, you should see the agent you selected.
6. In the Select the target agents table, select the check boxes for agents that should inherit settings from the source agent.
7. Click **Apply**.
8. When prompted for confirmation, click **Yes**.

Reviewing Rule Definitions

If you want to review the conditions of a rule, open the rule in the Rule Management dashboard.

IMPORTANT: Avoid editing Foglight for SQL Server rules in the Rule Management dashboard. These rules may be modified during regular software updates and your edits will be lost. Always use the alarm templates to modify rules.

You can create user-defined rules from the Rule Management dashboard. If you want to modify a rule, we recommend copying the rule and creating a user-defined rule. User-defined rules need to be managed from the Rule Management dashboard; these rules are not displayed in the Alarms view of the Databases Administration dashboard. For help creating rules, open the online help from the Rule Management dashboard.

To open the Rule Management dashboard:

1. On the navigation panel, under **Homes**, click **Administration**.
2. In the Administration dashboard, click **Rules**.
3. Type **DBSS** in the Search field to see the list of predefined rules for SQL Server databases.

   The Foglight for SQL Server rules are displayed. From here, you can review threshold values, alarm counts, and descriptions.
4. To see the full rule definition, click a rule and then click **View and Edit**.
5. In the Rule Detail dialog box, click **Rule Editor**.
6. When you are done your review, click Rule Management in the bread crumbs to return to the dialog box.
7. Click **Cancel** to avoid changing the rule unintentionally.

Defining Data Collection and Storage Options

The Foglight for SQL Server's agent collects and stores data at all times, even when the browser window that displays the data is not active. Use the Collection Frequencies view to specify:

- Which collections are sampled and stored
- The data collection values when sampling is carried out in offline, online, and real-time frequency modes
- The collection frequency

  When a user is currently focusing on a screen, the sampling frequency for all of the collections associated with this screen switches to Real-Time. The collection frequency determines the sampling frequencies of the other collections (that is, collections that are not running in Real-Time mode).

  The available collection frequencies are as follows:

  - Low — all collections are running in Offline mode, regardless of whether a client is connected.
  - Normal — the collections' running mode (Online/Offline/Real-time) adjusts dynamically to the client's connection status (disconnected/connected/focusing on a screen).
  - High — all collections are running in Online mode, regardless of whether a client is connected.
- The Query timeout for on-demand collections.

  This setting defines the number of seconds that a query for on-demand collections can run before it times out.

  On-demand collections are collections whose data is retrieved not by predefined time intervals but upon entering a screen or clicking a button.

  The default setting of this parameter is 60 seconds, but it can be modified by clicking the number that indicates the parameter's value in the field *Query timeout for on-demand collections:*.

NOTE: This section displays only SQL Server-related collections. For details about Windows-related collections, see Configuring Performance Counters on page 157.

Table 67. The *Collections* table includes the following columns:

Column	Description
Collection Enabled	Indicates whether the selected collections are sampled and stored.
Collection Name	The name of the collection. This list is sorted in an alphabetical order.
Offline Frequency (Sec)	Allows defining the collection interval, in seconds, in offline mode (for example, 300). Offline frequency refers to the longest interval possible for sampling the monitored instance.
Online Frequency (Sec)	Allows defining the collection interval, in seconds, in online mode (for example, 60).
Real-time Frequency (Sec)	Allows defining the collection interval, in seconds, in real-time mode (for example, 20). Only one collection can be sampled at real-time frequency in any given moment.
Query Timeout (Sec)	The amount of time, in seconds, that elapses before the query times out.

IMPORTANT: When the browser window that displays Foglight for SQL Server is active, the collection frequency mode in the active screen (for example: the Indexes pane in the Databases drilldown) switches to the fastest frequency possible - once every 20 seconds.

To modify the values of a specific collection:

1. Select the collection's row in the table.
2. Click **Edit**.

   The *Edit the Collection* dialog box appears.
3. Select whether to enable the collection and storage of the selected collection.
4. Set the collection interval, in seconds, in offline frequency mode (if available).
5. Set the collection interval, in seconds, in online frequency mode (if available).
6. Set the collection interval, in seconds, in real-time frequency mode.
7. Set the query timeout, in seconds.
8. Click **Set** to apply these settings or **Cancel** to reject them.
9. If no more editing is necessary, save the changes before switching to another screen. For details, see Reviewing the Administration Settings on page 144.

Defining Error Log Filtering

The Error Log Filtering view allows selecting which error logs generated by the SQL Server database are to be displayed in the Logs drilldown. The error log alarm increments the error log count for each error log entry that matches one of the strings listed in this view.

Foglight for SQL Server provides a default list of error logs enabled in the scanning, and allows adding, modifying, or disabling error logs from the list.

The settings defined using the Error Log Filtering view affect the following alarms: SQL Agent Error Log Summary, SQL Server Error Log Summary, SQL Agent Log Informational Error, and SQL Server Log Informational Error.

For details, see the Alarms chapter of the Foglight for SQL Server Reference Guide.

The Error Log Filtering view allows defining the following settings:

- Minimal severity for invoking summary alarms — can be either turned off or set to one of the defined severity values, that is: Warning, Critical, or Fatal.

  NOTE: By default, the value of this parameter is Critical.

- Minimal severity for invoking alarms — can be either turned off (the default setting) or set to one of the defined severity values: Warning, Critical, or Fatal.

For all SQL Server versions, the Error Log Filtering view allows defining the following settings:

- Display under a pre-defined name in the SQL Agent Error Logs and SQL Server Error Logs panels — using the **Match List** pane. This pane contains a default list of expressions within the alert logs that are to be retrieved and displayed in the Alert Log panel under a pre-defined name, category, and severity.
- Exclusion from the SQL Agent Error Logs and SQL Server Error Logs panels display — using the **Ignore List** pane, which contains a default list of alerts that are to be excluded when setting the alert logs display.

  IMPORTANT: Only messages that are explicitly defined in the **Ignore List** will not be displayed. Messages that were not added to either the Match or Ignore lists appear under name *Other*, type *SQL Server errors messages* and severity *Informational*. Therefore, ensure that messages that need not be displayed are added to the Ignore List.

Both the Match List and the Ignore List panes can be customized by adding, editing, or removing alert logs. Each filter can be enabled or disabled separately by clicking **Edit** and selecting or clearing the *Enabled* check box. Alternatively, to enable or disable all of the filters, click the **Enable All** or **Disable All** button.

To add an error to the Match List:

1 Click **Add**.

  The dialog box *Add Filter* appears.

  Use this dialog box to assign a name to the filter. Alert filters are enabled automatically upon addition. To disable a filter, use the **Edit** button.

2 Click **Add** to save your settings.

  Each newly added alert filter is enabled by default. To disable the filter, click **Edit** and then clear the *Enabled* check box.

The entire list of filters displayed on the view can be enabled or disabled by selecting the **Enable All** or Disable **All buttons** at the bottom of the screen. Enabling a single filter requires editing it.

To edit a filter:

1 Select the requested filter.

2 Click **Edit**.

  The *Edit Filter* dialog box appears. Ensure that the *Enable Filter* check box is selected (default).

3 To change the filter's name, edit the text in the Error Log Filter text field.

4 Click **Set**.

5 If another filter should be edited, repeat Step 1 to Step 4. If no more editing is necessary, or if all additional editing operations were carried out, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Configuring Performance Counters

The Performance Counters view allows configuring user-defined performance counters and their Unit of Measurement/Indicator. The counters created in this view are accessible using the User Metrics drilldown.

Each user-defined performance counter is collected and plotted over the specified time range in the User Metrics drilldown. The User Metrics drilldown is used only for displaying the user-defined performance counters; any creation or management operation of these counters is carried out using the Performance Counters view.

Use this view to add user-defined collections of performance counters to a specified agent by selecting them from the complete list of available performance counters. The user-defined counters value refers to raw data, which is derived directly from the counter provider.

To retrieve performance counters for a specific agent:

1 Select an agent from the list.

2 Click **Add Counters** at the bottom of the screen.

The *Retrieving Performance Counters* progress bar appears.

Upon successful completion of this operation, the *Add Counters* dialog box appears.

3 Select the requested counters. For each counter select the matching instance and unit.

4 Click **Add**.

5 After adding all of the requested counters, click **Close** to exit the Add Counters dialog box.

All of the newly added performance counters now appear in the table.

To delete unwanted counters, select the requested counter and click **Remove**. To add more counters, repeat Step 2 to Step 5.

6 After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Setting Options for Displaying Data in the Buffer Cache

The Buffer Cache view allows configuring the default retrieval settings for Buffer Cache panel, accessed through the Memory drilldown.

A lower buffer cache hit rate can be resolved by investigating the objects that consume the largest amount of cache size.

Use this view to configure the number of objects to be displayed in the table and their sorting properties.

Table 68. Click **Edit** in the *Buffer Cache Settings* section to edit the following parameters.

Parameter	Description
Top N Buffer Cache Objects by Cached MB	Defines the maximum number of objects to be displayed in the Buffer Cache panel (default: 20).
Order direction	Defines whether the order of the items for retrieval is ascending or descending.
Order by	Defines the parameter by which the display is to be ordered.

After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Setting Options for Displaying Data in the Plan Cache

The Plan Cache view allows configuring the default display settings for the SQL Server's plan cache panel, accessible using the **Memory** drilldown.

Use this view to filter the Plan Cache panel's display, by setting criteria such as which object type is displayed, as well as the default number of records displayed and the sorting method.

Table 69. Click **Edit** in the *Plan Cache Settings* section to edit the following parameters.

Parameter	Description
Show Adhoc objects	Defines whether to display Adhoc SQL plans.
Show system objects	Defines whether to display SQL Server system objects.
Filter Database Name	Defines whether to display only objects that reside on certain databases. When setting this parameter, the character % can be used as a wild card. For example, to display objects from all of the databases that begin with *Quest* (Questdatabase, QuestWorkDatabase, and so on), enter *Quest%*.
Top N Records	Defines the maximum number of objects to be displayed in the Plan Cache panel (default: 20).
Filter Object Name	Defines whether to retrieve only certain objects. When setting this parameter, the character % can be used as a wild card.
Order by	Defines the criterion for determining the data display order. The available criteria are as follows: • *Database Name* (Default) — the name of the database • *Schema Name* — the name of the schema • *Object Name* — the name of the object • *Object Type* — the object type • *Use Count* (Default) — the number of times this cache object has been used since inception • *Ref Count* — the number of other cache objects that reference this cache object • *Cache Object Type* — the type of the cache object • *SQL Bytes* — the size of the text • *SQL Text* — the SQL text • *Used Date Format* — the date format used by the object • *Used Language* — the language format used by the object • *MB* — the amount of space in the plan cache that is allocated to this object • *% from Cache* — the percentage of cache used by the object • *Used MB* — the size, in megabytes, used by the object type
Order direction	Defines whether the display would be carried out in ascending or descending order.

After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Setting Options for Displaying Data in the Locks Panel

The Locks view allows setting parameters for displaying data in the SQL Server's Locks panel, under the **SQL Activity** drilldown, as well as configuring the minimal duration, in seconds, which a block should reach or exceed in order to be collected and displayed in the Blocking (History) panel.

Table 70. The parameters that can when clicking **Edit** on this view are as follows:

Parameter	Description
Filter Database Name	Defines whether to display only objects that reside on certain databases. When setting this parameter, the character % can be used as a wild card. For example, to display objects from all of the databases that begin with *Quest* (Questdatabase, QuestWorkDatabase, and so on), enter *Quest%*.
Filter Object Name	Defines whether to display only certain objects. When setting this parameter, the character % can be used as a wild card.
Show System Objects	Defines whether to display SQL Server system objects.
Show Database Shared Locks	Defines whether to display shared locks.
Show Intent Locks	Defines whether to display Intent locks.
Show TempDB Locks	Defines whether to display locks on temp DB objects.
Blocking History section	
The minimal duration (seconds) for collecting a block in the Blocking (History) panel is:	Defines the minimal duration, in seconds, which a block should reach or exceed in order to be collected. All blocks that meet this criterion are displayed in the lower section of the **SQL Activity** > **Blocking (History)** panel**.**

After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Defining Retention Policies

The Retention Policies view allows defining the requested time range for which each of the metric collections, which are defined in the Collection Frequencies view, are to be kept.

Data can be retained for brief, moderate, or long periods, by selecting one of the following options:

- Short — retains data up to one month
- Medium — retains data up to three months
- Long — retains data up to one year

To modify the retention policy:

1. Click the text that displays the current retention policy.

   The dialog box *Edit the Retention Policy* appears.
2. Select the requested retention policy scheme from the list.
3. Click **Set** to apply the selected setting or **Cancel** to reject the setting.
4. After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Defining the Collection and Display of Top SQL Statements

The Top SQL Statements pane allows configuring the requested settings for top SQL statements, that is, statements that generated the maximal workload, as defined by the criterion by which the selected SQL statements are sorted. Use this pane to define the criterion of top SQL statements, as well as the maximum number of such statements to be retrieved and to be displayed. Top SQL statements are displayed on the **SQL Activity** > **Top SQL Statements** panel.

To define the settings for collecting and displaying SQL statements:

1 Click **Edit**.

The *Edit Top SQL Statements Settings* dialog box appears.

2 Use the *Maximum number of displayed SQL statements* field to enter the maximum number of SQL statements that are to be displayed in the Top SQL Statements table.

3 Use the *Maximum number of retrieved SQL statements* field to enter the number of SQL statements that generated the maximal workload and are to be saved to the Top SQL statements collection.

4 Use the *Sort the collected SQL statements by* list to select the field by which the list is to be sorted.

The possible values are as follows:

- CPU Time — the total CPU time consumed for carrying out the SQL statement executions
- Elapsed Time — the total time consumed for carrying out the SQL statement executions
- Executions — the number of times the SQL script executed for a unique SQL handle

5 Use the *Maximum size of short SQL statement* field to type the maximum number of characters for the short SQL text.

6 Click **Set** to save these settings.

7 After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Defining the Collection of Database Indexes

The Database Indexes view allows configuring the requested settings for collecting and displaying data under the **Database > Indexes** pane.

To define the settings for collecting and displaying database indexes:

1 Click **Edit**.

The *Edit Database Indexes Settings* dialog box appears.

2 Use the *Maximum number of collected indexes* field to type the maximum number of database indexes that are to be retrieved by the agent.

3 Use the *Sort the collected indexes by* list to select the field by which the list is to be sorted.

The possible values are as follows:

- Rows — number of rows in each index
- Index MB — the size of the index in megabytes

4 Use the *Collect indexes of type* field to select which types of indexes the agent collects.

The possible types are as follows:

- All

- Clustered
- Non-clustered
- Heap

5 Click **Set** to save these settings.

6 After carrying out all of the requested changes, save the changes before switching to another view. For details, see Reviewing the Administration Settings on page 144.

Configuring User-defined Collections

The User-defined Collections view in the Databases Administration dashboard allows adding user-defined collections to all of the currently selected agents, to provide for queries not included in Foglight for SQL Server.

IMPORTANT: Agents must be enabled for user-defined collections. If one or more of the selected agents is not enabled for such collections, this view allows enabling them. To disable or modify the credentials of a currently enabled agent, go to the Connection Details view and click the agent to edit it. For details, see Step 10 on page 146.

After collections are added, this view displays all of the user-defined collections for all of the agents; for example, if a collection was added to 12 agents during its addition, the view will display 12 rows, showing the collection for each agent.

This view can also be used for configuring the sampling frequency for each collection.

The available sampling frequencies are as follows:

- Real-Time - When a user is currently focusing on a screen, the sampling frequency for all of the collections associated with this screen switches to Real-Time.
- Online - When at least one user is connected, the sampling frequency for all of the collections that are not currently running at Real-Time frequency switches to Online.
- Offline - when no user is currently connected to the application

To add user-defined collections:

1 Click **Add**.

The User-defined Collections screen appears.

IMPORTANT: If one or more of the selected agents is not enabled for user-defined collections, a dialog box appears to notify this issue, displaying a table of the currently disabled collections. Use this dialog box to enable all agents for user-defined collections. To enable only part of these agents, exit the Databases Administration dashboard and select only these agents that are currently enabled or need to be enabled for user-defined collections before entering again the Databases Administration dashboard.

2 Type a name in the *Collection Name* field.

3 Enter a brief description of the collection in the *Collection description* field (optional).

4 Paste the query's SQL text in the Query Text field.

5 Type a value, in seconds, in the Query Timeout field.

6 Click **Verify**.

The collection is verified by running the query on each of the currently selected agents.

After the verification process is complete, the *Verification Results* pop-up appears, indicating whether the collection was verified successfully. In case the collection verification failed, the error message is displayed.

Table 71. If the verification succeeded on at least one agent, the collection details are displayed as below.

Section	Field	Description
Database	**ID**	The collections' ID
	Column Name	The name of the column
	Column Type	The field type, as retrieved by the query (String, Integer and so on)
Data Storage	**Display Name**	The column's display name
	Type	The topology type for storage purposes. This type can be one of the following: • String • Integer • Double • Boolean • Date **NOTE:** When the Frequently modified check box is selected (the default state), the field's change history is kept, including use of optional functionality such as use of the IntelliProfile mechanism and aggregation type selection. Clearing this check box is recommended only for fields whose values change infrequently, such as IP address of a specific host, as storing the change history of such fields is highly CPU-intensive and may degrade the FMS performance.
	Unit of Measurement/Indicator	The metric's Unit of Measurement/Indicator. The possible measurement unit values are: Percent, Count, Millisecond, Second, Minute, Hour, Day, Byte, Kilobyte, Megabyte and Gigabyte.
	Aggregation	Allows defining the value that is displayed in this metric, out of several values that were returned in the relevant time range. The available aggregation values are: • As is — value of the last sample taken during the selected time range • Sum — summarized value of all samples taken during the selected time range • Average — average value of all samples taken during the selected time range
	Is Key	Indicates whether the field is the query's key for retrieval. **NOTE:** Fields that are indicated as keys should correspond with the database result set unique values. Selecting a field which has frequently changing and repeating results as key is allowed, but may result in unexpected behavior from the defined collection.

Except for the Column Name and Column Type fields, whose values are retrieved by the query and cannot be changed, all other fields can be edited by clicking any of them.

The Edit Collection Properties dialog box appears, allowing you to edit the values of the following parameters:

- Display name
- Type
- Unit of Measurement/Indicator
- Aggregation

In addition, the collection's sampling frequencies are displayed on the table at the bottom of the view, and can be edited by clicking any of them.

7 Ensure that all settings are appropriate, and click **OK** to finish the collection creation process.

The newly created collection now appears on the table.

IMPORTANT: After adding the requested user-defined collections, they can only be deleted or cloned to other agents. If one or more queries need to be modified, delete them and create new ones.

Administering SQL Performance Investigator

The SQL Performance Investigator view in the Administration dashboard allows you to enable and disable SQL PI monitoring for selected agents. In addition, you can start and stop the collection of change tracking data.

In the SQL PI view, select one or more agents to enable or disable or for which to modify the change tracking status.

Configuring the On-demand Data Settings

The On-demand Data Settings view in the Databases Administration dashboard allows you to define a port for each Foglight Agent Manager (FglAM) to be used for retrieving data and integrating with SQL Performance Investigator.

To configure a port for the collection of on-demand data:

1 Select the check box near the names of the requested Foglight Agent Managers.

2 Click **Set port**.

The *Set On-demand Data Port dialog* box appears.

3 Type the name of the requested port.

4 Click **Set**.

5 Click **Validate connectivity**.

The Validate connectivity progress bar appears. At the end of the validation process, the connection status appears on the Status column.

If the connection failed, take the requested correction measures and try again.

IMPORTANT: Changes to the On-demand Data Settings view take effect immediately and do not need to be saved.

Configuring the Database to be Excluded

Monitoring all of the databases within all of the agents can unnecessarily load the system, as not all databases require such monitoring, due to being either non-mission critical or less significant. The Exclude Databases from Monitoring view allows excluding such databases from monitoring. Use this view to select the agents from which databases are to be excluded from monitoring, and then specify, either manually or by selecting from a list, which databases to exclude.

The On-demand Data Settings view in the Databases Administration dashboard allows defining a port for each Foglight Agent Manager (FglAM) to be used for retrieving data and integrating with Foglight SQL PI.

To exclude databases from monitoring:

1 Click **Exclude databases**.

> **NOTE:** Selecting a specific agent from which databases are to be excluded is done using the dialog box that opens upon clicking this button.

The dialog box Specify Databases for Exclusion appears.

2 Use the list *Select specific agent or all agents* to select an agent from which databases are to be excluded. Alternatively, select the option *All selected agents*.

3 Specify which databases to exclude from monitoring, either by adding their names manually or by selecting them from the list and clicking to move them to the list of excluded agents.

4 Click **OK**.

Reviewing Foglight for SQL Server Alarms

Alarms are the warnings that Foglight for SQL Server raises when a metric falls outside its “normal” range of values, which is defined by setting thresholds and severities for the metric within the Metric editor. A new alarm is raised whenever the severity for a metric changes. When the severity returns to normal, the alarm is canceled.

> **IMPORTANT:** Only several alarm types are invoked for instances monitored using the vFoglight for SQL Server Add-on mode. For details, see Viewing Data Displayed on vmExplorer on page 180.

> **NOTE:** For a complete list of alarms invoked by Foglight for SQL Server, including replication-related alarms, see Foglight for SQL Server Alarms in the *Foglight for SQL Server Reference Guide*.

Alarms Displayed in the Sessions Pane

Several alarms can be investigated using the home page’s **Sessions** pane and network packet flows, as follows:

- Response Time Alarm on page 165
- Packet Errors Alarm on page 166

Response Time Alarm

The Response Time alarm becomes active when the execution time of the Response Time SQL exceeds a threshold.

Response time is the full time (in milliseconds) it has taken a query (select 1, by default) to get from the application to SQL Server and back. Every time a real-time sampled interval starts (by default: 20 seconds), a query is sent and its response time value is displayed. Any value higher than 20 ms may indicate a performance issue, which should then be investigated to detect its source, identify the possible bottleneck, and take correcting measures.

The Response Time SQL is a user-defined Transact-SQL batch that can be used for indicating application response time.

To change the query used for determining response time, edit the Response time section in the Agent properties.

> **NOTE:** Only members of the Foglight Administrator group can change the SQL query used for measuring SQL Server response time.

Packet Errors Alarm

The Packet Errors alarm becomes active when the rate at which SQL Server is encountering network packet errors exceeds a threshold.

When this alarm is fired, investigate what is causing the packet errors on the network.

Alarms Displayed in the SQL Processes Panel

Several alarms can be investigated using the **SQL Processes** panel, as follows:

- Blocking Alarm on page 166
- Deadlocks Alarm on page 166
- Recompiles Alarm on page 167
- Error Log Alarm on page 168

Blocking Alarm

The Blocking alarm is raised when at least one SQL Server session is waiting on a lock held by another session. The waiting user is said to be “blocked” by the one holding the lock, and waits until one of the following scenarios realizes:

- The blocking user commits or rolls back, and therefore frees up the resource being waited on
- The blocked user's application timeout expires
- A deadlock happens

Following any of these scenarios, the blocked command is cancelled.

Excessive blocking can be a major cause of poor application performance, as users of an application often do not realize they are waiting on a lock held by another user. From their point of view, it often seems like their application has stopped responding.

When this alarm occurs, look at:

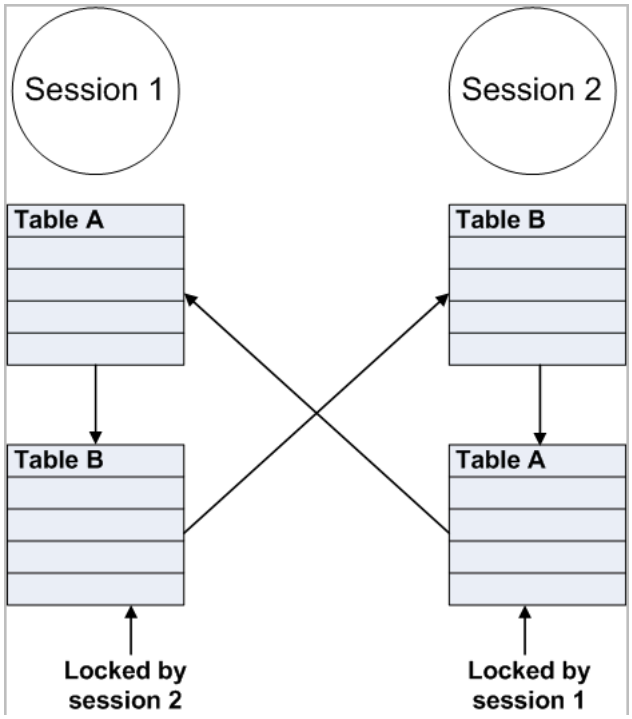
- The **Blocking** panel on the **SQL Activity** drilldown, to see who is blocking whom, and what resources are involved (for example, database and table names). In the case of multiple blocks, where blocked sessions are also blocking others, this panel displays the sessions at the top of the tree (those that do not have a “parent” in the tree). These sessions, by being at the head of the blocking chain, are the root blockers. This panel also shows how many sessions were blocked over time.
- The **Sessions** panel on the **SQL Activity** drilldown, to view the most recent SQL for the sessions involved in the blocking. This can help track down sub-optimal SQL that may contribute to the locking problem.
- The **Locks** panel on the **SQL Activity** drilldown, to view all locks in the system. This panel displays all SQL Server locks currently granted or requested.

Deadlocks Alarm

The Deadlocks alarm becomes active when the number of deadlocks per second exceeds a threshold.

A Deadlock occurs when multiple SQL Server sessions request conflicting locks in such a way that two locks are blocked by one another.

Figure 33. The following scenario details the simplest example of a deadlock, which involves only two sessions:



- Session 1 carries out a transaction that involves updating Table A
- Session 2 carries out a transaction that involves updating Table B
- Session 1 tries to update Table B, but needs to wait because the row is locked by Session 2
- Session 2 tries to update Table A, but needs to wait because the row is locked by Session 1

In this scenario, Session 1 is waiting on a lock held by Session 2, and Session 2 is waiting on one held by Session 1. In this example, to prevent both sessions from waiting forever, SQL Server enables only one session to continue and aborts the other session, by killing its connections and rolling back its updates. The user whose session is rolled back often receives an error message.

This scenario is relatively simple to resolve. However, deadlocks can involve many more sessions, which make it extremely difficult to track down the root cause of all the trouble. When this alarm occurs, review the **Locks** panel on the **SQL Activity** drilldown, to view the **Deadlocks/sec** counter on the **Lock Types** chart. This counter displays the frequency of deadlocks in the currently monitored system.

Recompiles Alarm

The Recompiles alarm becomes active when the ratio between the number of recompiles and the total number of compiles exceeds a threshold. This highlights when more than a certain percentage of compilations are due to run-time recompiles.

A compile can be a relatively time-consuming, CPU-intensive process, especially when the query or stored procedure is large or complex. For this reason, SQL Server stores execution plans in the Plan Cache, so that they would satisfy future I/O requests, thereby saving the need for physical reads from the disk.

A Recompile takes place when SQL Server Optimizer determines that the execution plan for a stored procedure that is currently executing may no longer be the optimal plan. SQL Server pauses the query execution and compiles the stored procedure again. This not only slows down the process that is executing the procedure, but adds extra CPU load on the server.

When many recompiles take place, the SQL Server's CPU can become overloaded, thereby slowing down everything running on that computer. Therefore, it is advisable to keep the number of recompiles as low as possible.

A stored procedure can be recompiled by the SQL Server for various reasons, the most common of which are as follows:

- Schema changes to any of the referenced objects, including adding or dropping constraints, defaults, or rules.
- A sufficient percentage of data changes in a table that is referenced by the stored procedure.
- Stored procedures performing certain operations on temporary tables.
- The use of the `WITH RECOMPILE` clause in the *CREATE PROCEDURE* or *EXECUTE* statement.
- Running *sp_recompile* against any table referenced in the stored procedure.
- High server activity causing the plan to be aged out of cache.
- Using different language and date formats.

To handle the Recompiles alarm:

- On the **SQL Activity** drilldown, click **SQL Instance Summary** and view the **Call Rates** graph to determine if this is a persistent problem. Consistently high percentage of the Re-Compiles rate within the total Compiles rate requires further investigation.

It is advisable to review the code of each of the stored procedures. Consider changing the stored procedures to remove coding practices that can cause recompiles.

Error Log Alarm

The Error Log alarm becomes active when Foglight for SQL Server, which scans the error logs at the configured frequency, detects messages that could indicate potential problems in the SQL Sever error log or SQL Agent error log.

When this alarm occurs, look at the Error Log drilldown to view the errors.

To select which error logs generated by the SQL Server database are displayed in the Error Log drilldown, use the Error Log Scanning view in the Databases Administration dashboard. For details, see Defining Error Log Filtering on page 156.

Alarms Displayed in the SQL Memory Panel

Several alarms can be investigated using the SQL Memory panel, as follows:

- Buffer Cache Hit Ratio Alarm on page 168
- Free Buffers Alarm on page 169
- Page Life Expectancy Alarm on page 169
- Procedure Cache Hit Ratio Alarm on page 170

Buffer Cache Hit Ratio Alarm

The Buffer Cache Hit Ratio alarm becomes active when the ratio of physical reads to logical reads falls below a threshold.

SQL Server holds recently accessed database pages in a memory area called the Buffer Cache. If an SQL process needs to access a database page, finding this page in the buffer cache saves SQL Server the need to read the page from disk, thereby significantly reducing the amount of disk I/O and, in most probability, speeding up queries.

Buffer Cache Hit Ratio is the ratio of logical reads to physical reads. It indicates the percentage of database page I/O requests that were satisfied from the Buffer Cache and therefore did not have to perform disk reads. This ratio measures how efficiently SQL Server is using the memory allocated to its buffer cache.

A low Buffer Cache hit rate indicates that SQL Server is finding fewer pages already in memory, and therefore has to perform more disk reads. This is often caused by either lack of SQL Server memory or use of inefficient SQL queries, which are accessing a very large number of pages in a non-sequential manner. The best figure varies from one application to another, but ideally this ratio should be above 90%.

To handle the Buffer Cache Hit Ratio alarm:

- Ensure that SQL Server is configured to use as much physical memory as possible. Check and, if necessary, alter the max server memory (MB) figure, which is displayed in the Configuration drilldown.
- If applications other than SQL Server are running on this computer, and the Total memory gauge on the main Foglight for SQL Server window shows that SQL Server is not using all of the memory it could, these applications could be taking memory away from SQL Server. For details, see Monitoring the SQL memory management on page 79.
- View the **Buffer Cache** panel on the **Memory** drilldown to see the largest objects in the Buffer Cache.
- Consider adding more physical RAM to the server.
- Use the **Sessions** panel on the **SQL Activity** drilldown to identify inefficient SQL queries. Look for currently active sessions that are generating a lot of I/O. Such sessions can be traced also by using the *Session Trace* pane on the **Sessions** panel.

Free Buffers Alarm

The Free Buffers alarm becomes active when the amount of SQL Server memory available for immediate reuse drops below a threshold.

The Lazy Writer Process periodically scans all SQL Server caches, and maintains a list of “free” pages that are available for immediate reuse.

When SQL Server needs a free memory page (for example, when reading a database page from disk into the buffer cache), and no free pages are immediately available, the connection needing the free page must wait while SQL Server makes buffers available. This results in slower performance. In the worst case, the connection has to wait while SQL Server writes a modified page out to disk, in order to make a free buffer.

This alarm does not always indicate a problem with SQL Server, especially if the alarm is not active for more than 10-20 seconds.

To handle the Free Buffers alarm:

- On the **Memory** drilldown, select the **Summary** panel. Check the Memory Areas chart to determine the amount of time for which the Free List has been very low. This alarm normally only indicates a problem if the Free List has been very low for more than a few minutes.
- If applications other than SQL Server are running on this computer, and the Total memory gauge on the main Foglight for SQL Server window shows that SQL Server is not using all of the memory it could, these applications could be taking memory away from SQL Server.
- Check other alarms on the home page. Factors such as stress on the memory manager and slow disk writes could all contribute to this alarm. Such factors appear as other alarms on the home page.
- Check the **Configuration** drilldown to view the currently set recovery interval parameter. Setting this too high can cause the Checkpoint process to run infrequently, which can in turn cause the Lazy Writer process to perform the majority of the I/O that the Checkpoint process normally does. This can keep the Lazy Writer so busy that it does not maintain the Free List efficiently.

Page Life Expectancy Alarm

The Page Life Expectancy alarm becomes active when the page life expectancy falls below a threshold.

Page life expectancy is the length of time in seconds that a database page will stay in the buffer cache without being accessed, before it is flushed out. Microsoft recommends keeping this value greater than five minutes (300 seconds).

Values smaller than 300 indicate that pages are being flushed out of the cache within a small period of time. The resulting lack of pages in the buffer cache requires SQL Server to carry out more disk reads, thereby degrading its performance.

This alarm is often invoked by memory shortage (either memory on the system or memory configured for SQL Server's use) or use of inefficient SQL queries, which are accessing a very large number of pages in a non-sequential manner.

To handle the Page Life Expectancy alarm:

- Ensure that SQL Server is configured to use as much physical memory as possible.
- If any applications other than SQL Server are running on this computer and the SQL Memory gauge on the main page shows that SQL Server is not using all the memory it could, then these applications could be taking memory away from SQL Server.
- Consider adding more physical RAM to the server.
- Identify inefficient SQL queries, by using the **Sessions** panel on the **SQL Activity** drilldown. Look for sessions that are currently active and generating a large number of I/O operations. This task can be carried out by using the **Session Trace** pane on the **Sessions** panel. For details, see Reviewing Session Trace details on page 88.

Procedure Cache Hit Ratio Alarm

The Procedure Cache Hit Ratio alarm is raised when the ratio between the number of times SQL Server looks for a plan in the plan Cache, and the number of times it finds the requested plan in the plan Cache, falls below a threshold.

A low plan cache hit rate indicates that SQL Server is finding fewer of the query execution plans it needs already in memory, and therefore has to perform more compiles. These extra compilations degrade SQL Server performance by causing extra CPU load.

To prevent this alarm being caused by adhoc SQL requests (which often produce non-reusable execution plans), Foglight for SQL Server removes adhoc plan statistics from this metric.

To handle the procedure cache hit ratio alarm:

- Check the Call Rates chart on the **SQL Instance Summary** panel of the SQL Activity drilldown for a high number of Re-Compiles. Follow the suggestions listed under the Recompiles alarm.
- On the **Memory** drilldown, select the **Plan Cache** panel and then use the Hit Rate and Use Rate counters on the Object Types chart to identify which types of objects are causing the problem.
- Ensure that SQL Server is configured to use as much physical memory as possible, as this alarm can be caused by an insufficient amount of memory being available for SQL Server to use.
- If any applications other than SQL Server are running on this computer, and the Total Memory gauge on the home page shows that SQL Server is not using all the memory it could, then these applications could be taking memory away from SQL Server.
- Consider adding more physical RAM to the server.
- View the Plan Cache panel on the Memory drilldown to see the list of objects in the cache.

Alarms Displayed in the Background Processes Panel

Several alarms can be investigated using the Background Processes panel, as detailed in the following sections:

- Cluster Server Down Alarm on page 171
- Log Shipping Alarm on page 171

- Non-preferred Cluster Node Alarm on page 171
- SQL Agent Alerts Alarm on page 172
- SQL Agent Job Failure Alarm on page 172
- SQL Server I/O Errors Per Second Alarm on page 172 Alarm
- Support Services Alarm on page 172
- Table Lock Escalation Alarm on page 173

Cluster Server Down Alarm

The Cluster Server Down alarm is raised when Foglight for SQL Server detects that at least one cluster node (server) is not currently running as part of the cluster.

When this alarm is active, take these measures:

- Check the **Cluster Services** panel of the **Support Services** drilldown to determine which cluster node is unavailable.
- Ensure that the Cluster Service is running on that server.
- Check the Windows event logs on the unavailable server to determine why it is not participating in the cluster.

Log Shipping Alarm

The log shipping alarm is invoked when the out-of-sync threshold has been exceeded for any log-shipping pairs, that is, when the time between the last backup of the source database and the restore in the target database has exceeded the allowed length specified. Because the log shipping operation comprises copy, backup and restore phases, the log shipping alarm can be invoked as a result of a failure to:

- Back up the primary SQL Server within the required period of time
- Copy the primary SQL server's backup database log to a secondary server within the required period of time
- Restore the secondary server's database from the backed-up database log within the required period of time.

When the alarm is raised, it is accompanied by a prompt to go to the **Log Shipping** panel of the **Support Services** drilldown. For details, see Tracking the Status of the Mirroring Operation on page 120.

Non-preferred Cluster Node Alarm

The Non-preferred Cluster Node alarm is raised when Foglight for SQL Server detects that SQL Server is not running on its preferred cluster node.

This alarm can be raised only when the currently connected SQL Server is running as part of a Microsoft Cluster Server (MSCS).

In a Windows cluster, each SQL Server instance belongs to a single cluster group. Preferred cluster nodes are allocated to each group. Normally, the group should run on these preferred cluster nodes.

To handle the Non-preferred Cluster Node alarm:

- View the Cluster Services page of the Support Services drilldown to see which SQL Server cluster group is not running on its preferred cluster node.
- Consider moving that cluster group to its preferred node.

SQL Agent Alerts Alarm

The SQL Agent Alerts alarm is activated when Foglight for SQL Server detects that at least one SQL Agent alert has occurred in the last few minutes.

To handle the SQL Agent Alerts alarm:

- On the *Support Services* drilldown, view the SQL Agent Alerts page to determine which alerts have occurred recently. This page displays the last occurrence time for each alert, and the alert history for the specified time range.
- Investigate the cause of the alert and take corrective action if necessary. For details, see chapter Using the Logs Drilldown on page 129.

SQL Agent Job Failure Alarm

The SQL Agent Job Failure alarm is activated when Foglight for SQL Server detects that at least one SQL Agent job has failed in the last few minutes.

To handle the SQL Agent Job Failure alarm:

- View the **SQL Agent Jobs** panel, on the **Support Services** drilldown, to determine which jobs have failed recently. Double-click any job to view the messages that it logged during its last run. This page displays the last run time and completion status of each job, as well as a graph showing which jobs ran recently, and the completion status for each run.

SQL Server I/O Errors Per Second Alarm

The SQL Server I/O Errors Per Second alarm is raised when I/O errors are encountered by SQL Server.

To handle the SQL Server I/O Errors Per Second alarm:

- View the SQL Error Log drilldown and look for messages indicating I/O problems.
- View the Windows Event Logs and look for messages relating to I/O problems.

The majority of I/O errors reported by SQL Server are caused by hardware failures, such as disk or controller failures.

Support Services Alarm

The Support Services alarm becomes active when any of SQL Server's supporting services are installed, but not active.

The services currently monitored are detailed in the following list.

- SQL Server Agent (SQLServerAgent)
- Distributed Transaction Coordinator (MSDTC)
- Microsoft OLAP/Analysis (MSSQLServerOLAPService)
- Full-Text Search (Microsoft Search)

To handle the Support Services alarm:

- Check the Service Status panel on the Support Services drilldown to see the status of all known SQL Server support services.
- Review messages in the SQL Error Log drilldown to determine the reason why the service is not running.
- Restart the affected service, if necessary.

Table Lock Escalation Alarm

The Table Lock Escalation alarm is raised when the number of times page locks are escalated to table locks per second exceeds a threshold.

Lock escalation is not necessarily a problem by itself; however, it can cause concurrency issues and lock conflicts and can be a major contributor to blocking problems.

In certain cases, this alarm may also indicate use of inefficient SQL queries, which leads to a large number of page locks instead of enforcing one table lock, and therefore forces the SQL Server to escalate the lock to a table lock.

Alarms Displayed in the Database Details Panel

Several alarms can be investigated using the Database Details panel of the Databases drilldown, as follows:

- Recent Backups Alarm on page 173
- Database Unavailable Alarm on page 173
- File Group Utilization Alarm on page 175
- Log Flush Wait Time Alarm on page 176
- Log Flush Wait Time Alarm on page 176
- Disk Queue Length Alarm on page 176

Recent Backups Alarm

The Recent Backups alarm becomes active when Foglight for SQL Server detects that any SQL Server database has not been backed up in the last few days.

To handle the recent backups alarm:

- On the Databases drilldown, check the Last Backup column in the Databases table for all databases.

Database Unavailable Alarm

The **Database Unavailable** alarm becomes active when Foglight for SQL Server detects that a SQL Server database is not available for reading. Users attempting to access an unavailable database receive an error message.

This alarm detects unusual database statuses, including **Suspect**, **Offline**, **Recovering, Loading, Restoring, Emergency Mode,** and others.

When this alarm occurs, you should:

- Determine which databases are unavailable. Check the Databases table on the **Databases** drilldown. The **Status** column shows which databases are unavailable.
- Take the action specified below for each unavailable database.

Some of the more common unavailable statuses are detailed in the following sections:

- Offline on page 174
- Loading or restoring on page 174
- Recovering on page 174
- Suspect on page 174

Offline

Setting databases offline can only be carried out manually, using the **sp_dboption** procedure. If any databases are Offline, consider using `sp_dboption` or `ALTER DATABASE` to bring the database online again.

Loading or restoring

Databases marked as **Loading** or **Restoring** are currently being restored by a `RESTORE DATABASE` or `RESTORE LOG` command. The database cannot be accessed by anyone while these commands are executed.

This status is also assigned to databases that have been restored using the `NORECOVERY` option. Specifying this parameter on a `RESTORE` statement notifies SQL Server that additional transaction logs need to be restored, and that no access to the database is permitted until these transactions are executed.

Check the **Sessions** panel on the **SQL Activity** drilldown for active sessions that are processing a `RESTORE` command (where the **Last Command** column contains **Restore**). If no sessions are processing a RESTORE command, the most likely reason for the database's unavailability is that the last restore was carried out using the `NORECOVERY` keyword.

Removing the **Loading/Restoring** status requires completing the `RESTORE` process. This can involve either waiting for the active `RESTORE` command to complete, or restoring the remaining transaction logs. The last transaction log should be restored without the `NORECOVERY` keyword. If the database is mirrored, a *Restoring* status is shown on the mirror.

Recovering

Databases are *Recovering* (or *InRecovery*) for a while when SQL Server is restarted, or the database is first set online. This is the status SQL Server uses for indicating that it is re-applying committed transactions, or removing uncommitted transactions after a SQL Server failure.

Normally, re-applying these transactions should take only a short time; however, if any long-running transactions were open when SQL Server ended abnormally, this procedure can take an extended period.

In some cases, it is advisable to bypass the SQL Server recovery process. For example, it would make much more sense to skip a lengthy recovery process when planning to drop the database as soon as the recovery process completes. For details on skipping the recovery process, see Bypassing SQL Server recovery on page 175.

CAUTION: Bypassing the recovery process can corrupt the database.

Suspect

Databases can be *Suspect* if they fail SQL Server's automatic recovery. This status most commonly appears after a SQL Server restart, when the automatic recovery process carried out during restart has failed. Databases can also be marked as Suspect when serious database corruption is detected.

The first measure that should be taken when a Suspect database is detected is to check the SQL Server error log, and look for error messages indicating recovery failure or database corruption. These messages should indicate the problem's cause.

To correct a suspect database, consider taking the following measures:

- Checking the SQL Server error log to determine why the database was made suspect.
- Ensuring that all database files are available. If any database file is unavailable when SQL Server attempts to open the database, the database is made suspect. Such a scenario can take place if a database file has been deleted or renamed while a SQL Server was down. It can also happen if another Windows process, such as Backup or Virus Scanning software, is using a database file when SQL Server tries to open it.

  In such a case, follow this procedure:

  a Wait for the database file to become available again.

  b Use the `sp_resetstatus` stored procedure (documented in Microsoft SQL Server's Books Online) to reset the database status.

  c Restart SQL Server to initiate recovery.

- If the Suspect status was caused by a full disk during recovery, free up disk space and use the `sp_resetstatus` stored procedure (documented in Microsoft SQL Server's Books Online) to reset the database status. SQL Server should then be restarted to initiate recovery.
- If the Suspect status was caused by a full disk during recovery, and it is not possible to free up space on existing database disks, add a new data or log file on a different disk that has free space available.
- Restore the database from the last full database backup, and then restore all transaction log backups taken since that point.

In most cases, a suspect database is best handled by restoring the database from the last good full database backup and transaction logs.

Using emergency mode

Emergency mode is a special status, which can be set on an individual database, thereby causing SQL Server to skip recovery for this specific database. In some cases, taking this measure can make the corrupt database available in order to extract data that cannot be retrieved in any other way.

Activating emergency mode causes SQL Server to skip the recovery of this database, thereby preventing the database being made suspect. However, the database may contain partially-complete transactions, and there may be inconsistencies between data and indexes (logical and physical corruptions). Do not carry out any database changes or updates when SQL Server is started in this way. Emergency Mode is documented at: http://support.microsoft.com/support/kb/articles/Q165/9/18.ASP.

Bypassing SQL Server recovery

Another high risk option to access a suspect database is to start SQL Server with Trace Flag 3608. This trace flag causes SQL Server to skip its automatic recovery process on ALL DATABASES when it starts. Again, this procedure may be sufficient for extracting data that cannot be retrieved in any other way.

- Use the `sp_resetstatus` stored procedure (documented in Microsoft SQL Server's Books Online) to reset the database status of any Suspect databases.
- Stop SQL Server, and then start it from a command line with Trace Flag 3608 and minimal startup (`sqlservr.exe -f -c -T3608`). This setting causes SQL Server to skip its automatic recovery at startup, thereby preventing the database from being made suspect. However, the database may contain partially-complete transactions, and there may be inconsistencies between data and indexes (logical and physical corruptions). Do not carry out any database changes or updates when SQL Server is started in this way.

With both *Emergency Mode* and *Bypassing SQL Server Recovery*, you may then be able to extract your data using *BCP.EXE* and/or script the database to get the latest database definitions. This can then be loaded into a new database using BCP.EXE or BULK INSERT. Be aware that the extracted data may not be complete.

File Group Utilization Alarm

The File Group Utilization alarm becomes active when a non-fixed size data file (that belongs to the file group) in any database is in danger of running out of space to grow.

This alarm is invoked whenever the space utilization percentage of a specific file group exceeds a predefined threshold value.

To resolve the data file growth limitation issue:

1. Under the **Databases** drilldown, click the **Data Files** panel.
2. Check for files with AutoGrow=Yes; files in danger of filling will have a low free percentage value (displayed on the Free Pct column).

Resolve this issue by freeing up disk space on the disk on which the file resides.

Example

The File Group Utilization alarm is raised when the following scenario takes place:

- All data files in the file group are 95% full, all these files have been configured with the AutoGrow option set to =Yes and, given the current growth increment, the data files have a limited number of growths remaining before all available disk space is consumed.

Log Flush Wait Time Alarm

The Log Flush Wait Time alarm becomes active when the duration of the last log flush for a database exceeds a threshold.

Because users make modifications to SQL Server databases, SQL Server records these changes in a memory structure called the Log Cache. Each SQL Server database has its own log cache.

When a user transaction is committed (either explicitly, by means of a COMMIT statement, or implicitly), SQL Server writes all changes from the Log Cache out to the log files on disk. This process is called a log flush. The user that issued the commit must wait until the log flush is complete before they can continue. If the log flush takes a long time, this degrades the user's response time.

Foglight for SQL Server checks the log flush wait time for the last log flush performed for each database. If a database has a slow log flush, and then has no update activity (and therefore no more log flushes) for a long time, Foglight for SQL Server continues to report this as an alarm until another log flush is performed for that database.

To handle this alarm:

- On the **Databases** drilldown, select the **Summary** panel to review the Log Flush Wait Time counter in the Database History graph. The database with the high graph values is the one experiencing the problem. If a database has a consistently high value that never changes, run SQL command CHECKPOINT on that database to force another log flush and check the value in Foglight for SQL Server again.
- Select the **Transaction Logs** panel on the **Databases** drilldown to find the disks on which the log for this database resides.
- Consider moving the log files to disks that support fast write activity (for example, a fast RAID controller with write-back caching enabled).
- Consider moving log files off RAID-5 devices as these are optimized for read activity, and log files generate mainly write activity.

Disk Queue Length Alarm

The Disk Queue Length alarm becomes active when the disk queue length of any disk exceeds a threshold. Sustained high disk queue length may indicate a disk subsystem bottleneck, and usually results in degraded I/O times.

Disk queue length is a Windows-based metric. Therefore, occurrence of the Disk Queue Length alarm does not necessarily indicate a problem with the SQL Server instance, and can be the result of I/O operations carried out by non-SQL Server processes. Nevertheless, SQL Server, as well as any other application running on the computer for which this alarm is raised, is affected by slower disk throughput.

To handle the Disk Queue Length alarm:

- On the **SQL Activity** drilldown, click the **SQL I/O Activity** panel and look at the *SQL Server Physical I/O* chart, to view whether SQL Server is generating high amounts of disk activity. This chart displays the rate (I/O per second) for each type of I/O that SQL Server is performing. If SQL Server is not generating a lot of I/O activity, the high disk queue length is most likely being caused by some other Windows process, or by Windows itself.
- On the **SQL Activity** drilldown, click the **Sessions** panel to see which SQL Server processes are executing at the time the alarm was raised, and the SQL currently being executed.
- Consider moving database files to faster disks. If you are not using hardware RAID, consider purchasing a RAID subsystem. If you are using RAID-5 for write-intensive files (such as Database Logs or heavily updated database files), consider moving to a faster RAID implementation (RAID-0 or RAID-10).

- In some cases, you can speed up all disk I/O by reviewing the RAID options on your RAID controllers. One example is to enable disk-write caching, as long as your disk subsystem is protected by battery backups or UPS.
- On the **SQL Activity** drilldown, click the **SQL I/O Activity** panel and look at the *SQL Server Physical I/O* chart, to view the Checkpoint statistic. If the Checkpoint process is generating a lot of I/O, review the Recovery Interval setting in the Configuration drilldown.

Creating Blackouts for Selected Instances

In certain scenarios, users may prefer not to receive alarms from one or more agents for a predefined period of time (for example, during the night hours or when a scheduled maintenance is carried out).

Creating a blackout for one or more databases within an instance requires defining the requested instance as a service, and then creating a blackout for this service.

To create a service:

1. Go to **Dashboards** > **Services** > **Service Builder**.
2. Click the sign near the title Add a New Category.

   The New Category Wizard dialog box appears.
3. Enter a name for the service in the Name field.
4. In the Tier list, select the User option.
5. Select the option **No** in the field that requests indicating whether Foglight should dynamically add and maintain the hosts for this service.
6. Click **Finish**.

   The newly created service appears on the table.
7. In the Actions column, click Add ( ) and select the command **Add components to this service**.

   The dialog box Add components to this service appears.
8. Click **Add specific component**.
9. Use the *From* list to select the option *Agents*.
10. Click **Search**.

    The list of agents appears on the table.
11. Select the boxes near the requested agents.
12. Click **Add Components**.
13. Repeat Step 7 to Step 8.
14. Use the From list to select the option *All Models*.
15. Click the + sign near the name DBSS_Data_Model, to expand this field.
16. Select the instances for which no alarms should be invoked.
17. Click **Add Components**.

To create a blackout for the newly created service:

1. Go to **Dashboards > Administration > Setup & Support > Blackouts**.
2. Click the option **Create a One-Time Blackout**.
3. In the next screen select the option **Suspend Alarms**.
4. In the screen *Choose Target Type*, select *Services* and click **Next**.

The screen *Choose Blackout Targets* appears.

5 Select the previously created service for which no alarms are to be invoked and click **Next**.

6 Use the next screen, *Specify Blackout Period*, to select the start and end times of the blackout period and click **Next**.

The screen *Configure Name and Reason* appears.

7 Select whether to leave the default name of the blackout or to enter a new name, as well as the reason for blackout.

8 Click **Finish** to complete the procedure.

The *Blackout Summary* screen appears, displaying the details of the newly created blackout.

Generating Reports

Foglight for SQL Server allows generating reports about various aspects of the selected instance performance. This chapter provides information on how to generate the various reports, as well as a brief description of each report.

NOTE: For detailed information regarding the use and configuration of reports, see *Foglight User Guide* > *Working with Reports*.

Generating Reports for a Foglight for SQL Server Instance

To generate reports for a selected instance:

1 Go to **Dashboards** > **Reports**.

2 Click **Generate a Report**.

The Generate Report dialog box appears.

3 Select **Templates By Module > Databases > SQL Server > Reports**.

4 Select the requested report.

5 Click **Next**.

6 Select the requested time range.

7 Select the instance for which the report will be generated.

8 Click **Next**.

9 Assign a name for the report.

10 Select the report format (PDF, Excel or XML).

11 Use the Email Recipients field to type the name of the report's recipients.

12 Click **Finish**.

13 Upon successful completion of the report generation process, the *Report Generated Confirmation* dialog box appears.

14 To view the report, click **Download Now**.

15 Select whether to save the file or to open it using the relevant program for the selected format.

Studying the Various Reports

Use the Reports dashboard to generate reports for Foglight for SQL Server at the following levels

- Instance-specific reports — require selecting an instance before proceeding to the report generation. For more information, see Reviewing Instance-specific Reports on page 179.

Reviewing Instance-specific Reports

The following reports are generated at the instance level:

- Configuration report — Displays the specified instance's most updated configuration
- Change Tracking report — Displays the specified instance’s changes
- Database Space Usage — Displays a breakdown of the database storage space usage at the host, instance, and database levels.
- Enterprise SQL Server Performance Summary — Displays various performance indicators for all currently monitored instances
- Enterprise Inventory and Availability — Displays various inventory information for all currently monitored instances
- Enterprise SQL Server Setup — Displays general setup details for all currently monitored instances
- Health Check Report — Displays various aspects of the specified instance health
- I/O Activity Report — Displays comprehensive analysis of I/O activity, including activity vs. I/O, wait for I/O, and buffer usage
- Memory Report — Provides a detailed report regarding memory usage, such as SQL Server memory vs. host RAM, buffer cache size, page life expectancy and plan cache hit rate
- SQL Server Executive Summary Report — Provides key decision-makers with a summary view of the SQL Server activity.
- Storage Report — Provides an overview of disk space usage, including top consuming databases and files breakdown storage usage
- Top DB Users — Displays the DB users that generated the maximal workload during the specified time range.
- Top SQL Batches — Displays the top-consuming SQL batches that were executed during the specified time range.

  NOTE: The maximum number of top SQL batches that can be generated is 10. Increasing the value to a number higher than 10 has no effect.

- Deadlocks Report — Display the deadlocks detected during a given time range.
- In-Memory OLTP (XTP) Summary Report — Displays information gathered for InMemory OLTP (XTP) Memory consumption, Databases and Waits.
- Disk Space Usage — Displays a breakdown of the disk storage space usage at the host and instance levels. on a selected instance.
- Top Disk Space Usage — Displays a breakdown of the disk storage space usage at the host and instance levels of top space consuming instances filtered by free space (% or MB).
- Workload Summary Report — Displays overall workload, using various performance indicators
- Storage Planning Report— Displays a prediction of storage growth using a linear regression method, based on the available historical data (up to 3 months).

Monitoring SQL Server instances on VMware servers

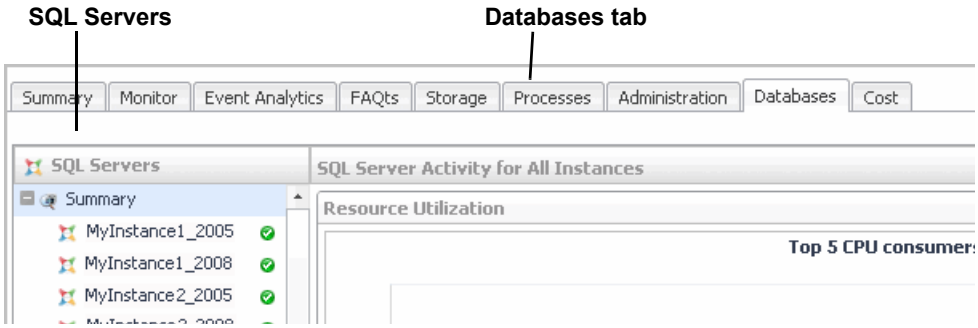
Users of Quest Foglight for Virtualization, Quest's solution for performance monitoring and management of virtual environments, can benefit from the SQL Server functionality, which provides light-weight monitoring of SQL Server instances running on VMware ESX servers.

This functionality assists database administrators, who need to investigate the share of SQL Server-related processes within the overall system workload.

Viewing Data Displayed on vmExplorer

The Databases pane on the vmExplorer dashboard (**Dashboards** > **VMware** > **vmExplorer** > **Databases**) is the main tool for investigating the share of the SQL Server-related processes on the virtual machine's overall system workload.

Figure 34. The Databases pane on the vmExplorer dashboard.



This pane allows viewing SQL Server activity in the modes described in the following sections:

- Reviewing SQL Server Activity for All Instances on page 180
- Reviewing SQL Server Activity for a Specific Instance on page 181

Reviewing SQL Server Activity for All Instances

The SQL Server Activity for All Instances view, displayed by clicking **Summary** under the SQL Servers section, displays summarized data of SQL Server activity for all instances residing on the selected Virtual Machine.

This screen contains the following sections:

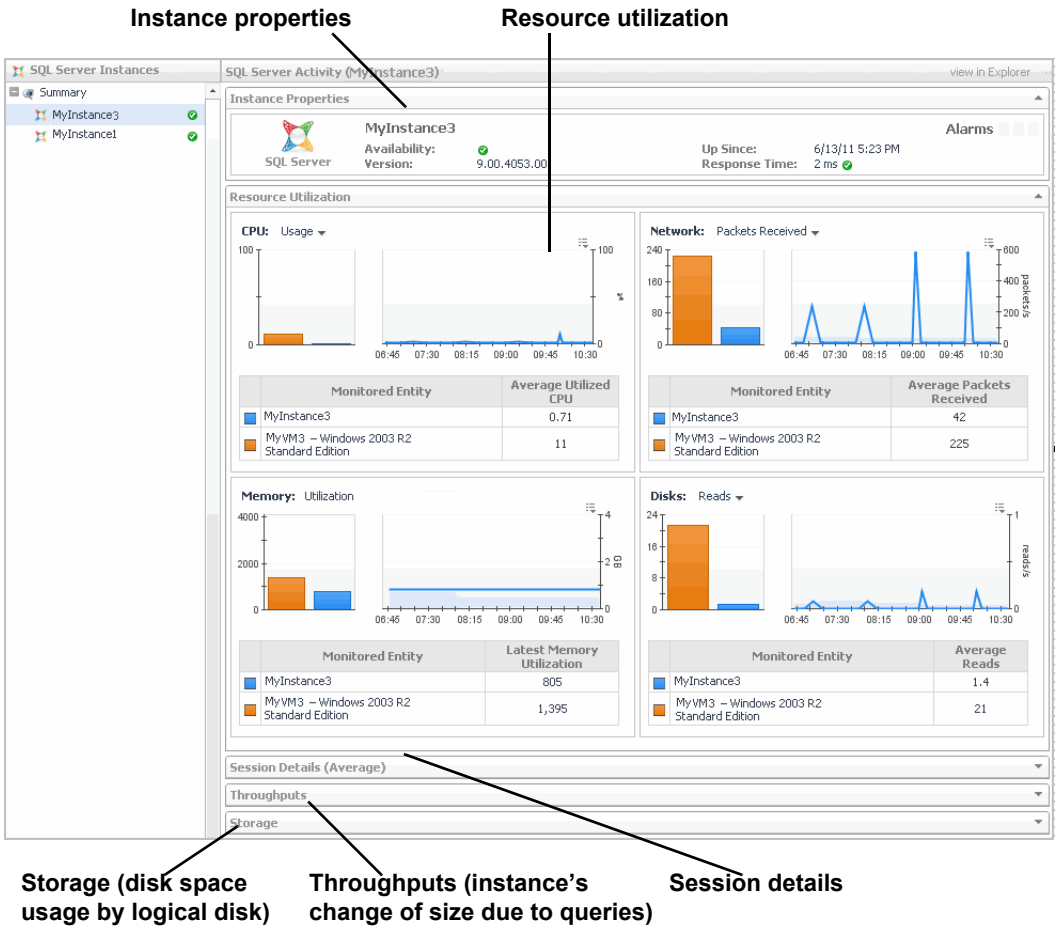
- Resource Utilization — displays data derived from VMware, showing the SQL Server instances, monitored in both monitoring modes, which had the greatest impact on the system performance:
  - Top 5 CPU consumers
  - Top 5 consumers of network resources
  - Top 5 memory consumers
  - Top 5 disk space consumers
- vFoglight for SQL Server Add-on related Alarms — displays all alarms invoked for instances monitoring in vFoglight for SQL Server Add-on mode. These alarms correspond, both in number and time range, to the alarms displayed on the Databases table and Quick View panel for instances monitored in this mode.
- SQL Server - FAQts — allows viewing the most activity-intensive instances in various parameters.

This section contains multiple questions regarding these issues, such as: “Which instances had the highest rate of I/O Reads?”. The answer to each of these questions is provided, in both a chart and a table format, using a popup displayed by clicking **Show Me**.

Reviewing SQL Server Activity for a Specific Instance

The SQL Server Activity view is displayed by clicking a specific instance under the SQL Servers section.

Figure 35. The SQL Server Activity view.



This view allows carrying out the tasks described in the following sections:

- Viewing instance properties on page 181
- Viewing resource utilization on page 182
- Viewing session details on page 182
- Viewing the instance’s change of size on page 183
- Viewing disk space usage by logical disk on page 183

Viewing instance properties

The Instance Properties section contains the following components:

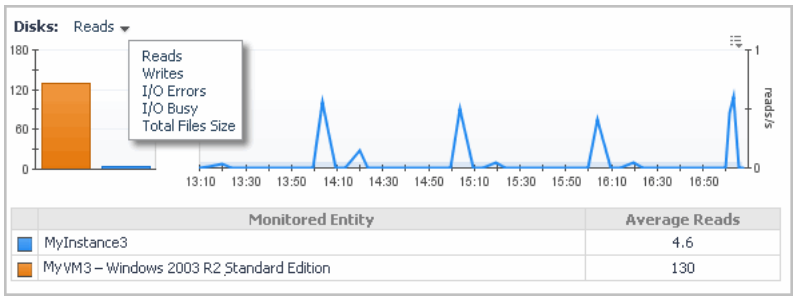
- Instance indicators — Availability, Version, Up Since, and Response Time.
- Alarms — the system-related alarms, derived from VMware, which were invoked for the selected instance during the specified time range.

Viewing resource utilization

The Resource Utilization section displays the instance's utilization of the following resource types: CPU, Network, Memory, and Disk I/O.

For each of the sections, whenever there is an equivalent metric in the Virtual Machine topology, a bar chart is displayed, comparing the average between the period values of the SQL Server instance and the Virtual Machine metrics. The chart displays the selected metric's values throughout the specified time range, and the table displays the metric's average value.

Figure 36. Resource utilization chart.



NOTE: For the Memory resource, the table displays the latest memory utilization.

Except Memory, the utilization of each resource category can be seen using several metrics, as listed below.

- For the CPU resource:
  - Usage (CPU Utilization)
  - Idle
- For the Network resource:
  - Packets received
  - Packets sent
  - Total packets
  - Packet errors
- For the Disks resource:
  - Reads
  - Writes
  - I/O errors
  - I/O busy
  - Total files size

Viewing session details

The Session Details (Average) section displays the following summarized data:

- Total Sessions — the average number of SQL Server processes, both user and system processes.
- Active Sessions — the number of non-system sessions that are actively processing in SQL Server or that are waiting on locks (blocked).
- Blocking Sessions — the average number of blocking processes that were collected during the specified time range.
- Blocked Sessions — the average number of blocked processes that were collected during the specified time range.

- Deadlocks — the average number of deadlock events that took place during the specified time range, plotted over time.

NOTE: Clicking the icon near each of the numerical values displays a popup with a chart that shows these values plotted over the specified time range. For instances monitored in Foglight for SQL Server mode, a link at the bottom of each popup allows further investigation using the relevant panel in the **SQL Activity** drilldown.

Viewing the instance's change of size

The throughputs section contains two bars, which display the instance's accumulated change of size caused by running SQL queries of the type selected from the Query Type list (Insert, Update, Delete, Select, or All) during the specified time range. The size change is shown using the following indicators:

- Number of megabytes
- Number of rows

NOTE: Clicking anywhere on the bar displays a popup. For instances monitored in Foglight for SQL Server mode, a link at the bottom of this popup allows further investigation using the **SQL Activity** > **I/O by File** panel.

Viewing disk space usage by logical disk

The Storage section displays, for each of the logical disks, storage-related data about the disk space allocation and utilization by the virtual machine and the SQL Server instance. Virtual machine-related data is retrieved using vFoglight, while SQL Server instance-related data is retrieved using Foglight for SQL Server. Data is displayed in a graphic (bar) and textual formats.

The **Disk Space Utilization** bar displays the space breakdown of the logical disk into the following categories:

- Database used space — disk space actually used by the selected instance.
- Database free space — disk space allocated to the selected instance but not used.
- Virtual machine used space — total disk space used by the virtual machine out of the entire disk space, excluding space allocated to the selected instance.
- Virtual Machine free space — total space allocated to the virtual machine but not used, excluding space allocated to the selected instance.

The *Virtual Machine* section displays the following data regarding the virtual machine's disk space utilization:

- Total space — total disk space allocated to the virtual machine.
- Free space — total disk space allocated but not used. This value is a summary of the database free space and the virtual machine free space.
- Used space — total disk space used by both the virtual machine and the selected instance.

The *Database* section displays the following data regarding the SQL Server instance's disk space utilization:

- Allocated Space — disk space allocated to the instance
- Used Space — disk space actually used by the instance

5

Reference

This chapter contains reference information about the following topics:

- SQL Performance Investigator Metrics
- Rules
- Collections and Metrics

SQL Performance Investigator Metrics

Active Time

Sum of all active wait events; equal to the session total activity within the specified time range.

All SQL Agents CPU Usage

CPU consumption by all of the SQL Agents running on the monitored host.

All SQL Agents Resident Memory Usage

The amount of physical memory (RAM) consumed by the monitored instance's SQL Agent service.

Availability

This metric indicates whether the SQL Server instance is available.

Average Physical I/O Operations

The average time spent on performing physical I/O operations (both read and write), out of the total active time during the specified time range.

This metric is calculated as follows:

(Page Reads + Page Writes)/Active Time

Average SQL Response Time

Average duration of the SQL statements that were executed during the specified time range.

This metric is calculated as follows: Ended Duration/Executions Ended

Backup Recovery Wait

Time spent by the various sessions waiting for backup/recovery tasks to complete.

Blocked Lock Requests

Number of lock requests that could not be satisfied immediately, causing the caller to block and wait before acquiring the lock.

Checkpoint Pages

The number of pages flushed by checkpoint or other operations that require all dirty pages to be flushed.

CLR Wait

Time spent by the statement waiting for CLR code execution to complete.

CLR, which stands for Common Language Runtime, introduced in SQL Server 2005, allows hosting of the Microsoft .NET common language runtime engine within SQL Server.

CPU Usage

Time spent (in seconds) by the various sessions consuming CPU cycles.

This reading is taken directly from the operating system, rather than SQL Server wait states.

CPU Wait

Wait Time (seconds) until the CPU resource is available. Time spent by the session waiting in the system's run queue for CPU cycles. The amount of time is dependent upon the number of concurrent processes and threads requesting CPU time. The metric value should be inspected in conjunction with the value of the Run Queue Length metric.

Cursor Synchronization Wait

The time spent by the various processes synchronizing information flow within cursors.

Cursors are extensions to result sets, which provide the mechanism for working with individual rows, or a small block of rows, in a table.

Because a cursor points to a currently selected set of records, a cursor can be used by only one connection at a time. However, the compiled plan to which the cursor is linked can be used simultaneously by multiple connections.

Database Replication Wait

Time spent by the various sessions waiting for replication synchronization events to complete.

Deferred Task Worker Wait

Time spent by the various sessions waiting for I/O requests; this wait indicates that either a large number of suspect pages have been encountered or the disk subsystem is overloaded.

Degree of Parallelism

Average number of SQL Server threads assigned to serve a SQL statement, for each parallel plan execution.

Disk Utilization

The percentage of time the busiest disk spent serving system-wide I/O requests.

This metric serves as a measure for the system I/O load. High values may indicate a device bottleneck.

DTC CPU Usage

The percentage of CPU consumption by the MS-DTC service.

DTC Resident Memory Usage

The amount (MB) of physical memory consumed by the MS-DTC service.

Distributed Transaction Wait

Time spent by the various sessions waiting for various distributed transaction events to complete.

This wait type can occur either because the MS-DTC service becomes unavailable, or because a task is waiting for the completion of an MS- DTC transaction’s commit.

Executions Ended

The average number of statements whose activity completed during the specified time range.

Executions Started

The average number of statements that started running during the specified time range.

External Procedures Wait

Time spent by the various sessions waiting for external procedures to end.

An external procedure, also sometimes referred to as an external routine, is a procedure stored in a dynamic link library (DLL) on Windows platform or shared library under UNIX. The external procedure is registered with the base language, and then invoked to perform special-purpose processing.

Because external procedures run in a process separated from the database instance, using these procedures ensures that any problems on the client side do not adversely affect the database.

Full Scans

Number of unrestricted full scans per second. These can be either base-table or full-index scans.

Full Text Search CPU Usage

Average CPU consumption by the Full Text Search service.

Full Text Search Resident Memory Usage

Average amount of physical memory consumed by the Full Text Search service.

Full Text Search Wait

Time spent by the various sessions waiting for full text synchronization operations.

Full text search is a Windows service that can optionally be installed with SQL Server. This service, which is also called MSSearch service, allows indexing of selected columns in a way that allows users to search any word, phrase, or derivative of any word contained in that column.

Full Text indexes are not updated in real time.

Free Buffer Wait

This wait event occurs when the session needs a free buffer, so it can bring a data block into the buffer cache, and is waiting for a buffer that is not dirty to become available. This can occur if DBWR is not writing dirty buffers to disk fast enough.

Hosted Components Wait

Time spent by the various sessions waiting for hosted components, such as, but not exclusively, CLR.

IO Bulk Load Wait

Time spent by the various sessions waiting for the completion of I/O operations required to carry out a bulk load I/O.

While bulk loading significantly reduces the amount of I/O activity required for loading large amounts of data to the database inside of a single transaction, this operations still takes a relatively long time to complete.

IO Completion Wait

Time spent by the various sessions waiting for I/O operations to complete.

IO Data Page Wait

Time spent by the various sessions waiting to latch a buffer for an I/O request; this wait is related to issues with the disk I/O subsystem.

IO Wait

Time spent waiting for disk input/output operations to complete.

Input/output (I/O) is one of the most expensive operations in a database system. SQL statements that are I/O intensive can monopolize memory and disk use and cause other database operations to compete for these resources.

Generally, I/O Wait is caused by poorly-tuned SQL queries or applications which generate a significant amount of logical I/O translating into excessive physical disk usage. In this case, SQL/application tuning can reduce the logical I/O- induced load. However, it could also be caused by poorly-configured disks or storage sub-systems.

Latch Buffer Wait

Time spent by the various sessions waiting to latch a buffer that is not a physical I/O request.

Latch Wait

Time spent by the session being blocked by a latch, waiting for it to be released.

Latches do not need to be locked for the duration of a transaction. They are low-overhead, short-term memory synchronization objects. They are used mostly to protect a row when queried for a connection.

Latch Savepoint Wait

Time spent by the various sessions waiting to synchronize commits to marked transactions.

Lazy Writes

The number of buffer pages that have been modified in the buffer cache, which the Lazy Writer process is flushing to disk.

A high value of lazy writes may indicate that SQL Server is running out of available space in the buffer pool cache.

Lock Wait

Time spent by the session being blocked, waiting for the blocking lock to be released.

Lock Bulk Update Wait

Time spent by the various sessions waiting to acquire bulk update locks.

Bulk update locks are applied by the database engine when multiple threads perform bulk copying of data into the same table. Applying these locks allows the threads to bulk load data at the same time into the same table, while preventing access to the table from other processes, which do not carry out data bulk loading.

Lock Exclusive Wait

Time spent by the various sessions waiting to acquire bulk update locks.

Lock Intent Wait

Time spent by the various sessions waiting to acquire intent locks.

Intent locks are locks used by the database in order to protect a resource that is found lower in the lock hierarchy from a shared (S) lock or exclusive (X) lock. Because these locks are acquired before a lock at the lower level, they signal intent to place locks at a lower level.

Lock Requests

Number of new locks and lock conversions requested from the lock manager.

Lock Schema Wait

Time spent by the various sessions waiting to acquire schema locks. Schema locks are used for preventing changes to the underlying tables structures while the query is compiled.

Lock Shared Wait

Time spent by the various sessions waiting to acquire shared locks.

Lock Update Wait

Time spent by the various sessions waiting to acquire a lock update.

Lock Wait

Time spent waiting for blocking locks to be released.

Log Buffer Wait

Time spent by the various sessions waiting for space in the log buffer or otherwise waiting for memory to be made available to write log records. Consistently high values of this metric can indicate the log devices' inability to keep up with the amount of log generated by the server.

Log Flushes

The Number of log flushes.

As users make modifications to SQL Server databases, SQL Server records these changes in a memory structure called the Log Cache. Each SQL Server database has its own log cache.

When a user transaction is committed (either explicitly using a COMMIT statement, or implicitly), SQL Server writes all changes from the Log Cache out to the log files on disk. This process is termed a log flush. The user that issued the commit must wait until the log flush is complete before they can continue. If the log flush takes a long time, this will degrade the user's response time.

Log Other Wait

Time spent by the various sessions waiting for miscellaneous logs.

Log Synchronization Wait

Time spent by the various sessions waiting for log space to be freed by log truncation, that is, deleting log records from the transaction log.

Even though log transaction is an automatic process, this process can be delayed as a result of several factors, such as long running transaction, a data backup taking place, or if no checkpoint took place since the last log truncation.

Log Wait

Time spent waiting by the various processes waiting for a log operation to complete.

Log Write Wait

Log write wait takes place when a task is waiting to write the contents of the log cache to the disk that stores the transaction log.

Memory Wait

Time spent waiting by the various processes waiting for the completion of a log operation.

Time spent by the various sessions waiting for outstanding HTTP connections to complete and exit.

Network IO Wait

Time spent by the various sessions waiting for network packets.

Network IPC Wait

Time spent by the various sessions waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.

Network Mirror Wait

Time spent by the various sessions waiting for database mirroring events to complete.

Network Wait

Network wait events occur when a session spends time waiting for messages to be sent or received over the network interface.

Network performance is measured in number of packets sent and received per second. It can be used just like disk statistics to detect overload or non-optimal performance within a network or a network interface.

Excessive network wait can result from either:

- Excessive network usage, originating in the application
- Physical issues, identifiable by network errors and network collisions

Non SQL Server CPU Usage

CPU consumption on the server by processes other than SQL Server.

Non SQL Resident Memory Usage

The amount of physical (resident) memory consumed on the server by processes other than SQL Server.

OLAP CPU Usage

CPU consumption of the Analysis Services processes.

OLAP Resident Memory Usage

The amount of physical (resident) memory consumed on the server by the Analysis Services processes.

OLEDB Provider Full Text Wait

Time spent by various processes waiting for the Microsoft Full Text Engine for SQL Server.

Other CPU Usage

CPU consumption of processes not related to SQL Server or associated services.

Other Miscellaneous Wait

Time spent by the various sessions waiting for miscellaneous database operations.

Other Wait

Time spent by the various sessions waiting for other database operations.

Overall CPU

Time spent by the various sessions either consuming or waiting for CPU cycles. This reading is an aggregation of CPU Usage and CPU Wait, which are both taken directly from the operating system (rather than MSSQL wait states).

Page Life Expectancy

Number of seconds a page will stay in the buffer pool without references. Low values of this metric require the buffer pool to frequently flush the page to the physical disk, thereby possibly indicating a memory issue. Such an issue should be investigated to see if it results from lack of physical memory, inefficient queries and so on.

Page Splits

Number of page splits taking place.

Page splitting occurs when an index or data page becomes full and then is split between the existing page and a newly allocated page.

Excess page splitting can lead to I/O performance issues, indicating a need to rebuild indexes and/or increase the indexes' fill factor.

Parallel Coordination Wait

The time spent by the various processes coordinating parallel query threads and exchanging data.

If the server on which SQL Server is running has multiple CPU cores, SQL Server can run a single query in parallel, using multiple threads. In addition to running user queries on multiple processors, SQL Server can also use multiple threads to build indexes. Typically, queries that make heavy use of CPU cycles are good candidates for parallel execution. For example, a query joining several large tables and sorting the output before returning it to the user is likely to benefit from a parallel execution plan.

Physical I/O

The amount of physical I/O operations (both read and write) carried out by SQL Server during the specified time range.

Physical Memory Used

The amount of physical memory, or RAM, being used by the server.

Physical Page Reads

The number of physical database page reads issued during the specified time range.

Physical Page Writes

The number of physical database page writes issued during the specified time range.

Probe Scans

The rate per second of probe scans. A probe scan is used for directly looking up rows in an index or base table.

Plan Cache Hit Rate

The Plan Cache is an area in the SQL Server memory, used for storing objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers.

The plan cache hit ratio indicates the ratio between physical accesses and logical accesses, that is, how much work (for example, I/O and compilations) is being saved by finding the required information already in the plan cache.

Range Scans

Number of qualified range scans through indexes.

Rec Ended Duration

Elapsed (gross) duration of the statements that ended during the current time interval.

Remote Provider Wait

The time spent by the various processes waiting for a remote OLEDB call to complete or DTC synchronization.

Run Queue Length

System average run queue.

The CPU run queue is a holding area for threads and processes that require the CPU when the CPU is busy serving other processes. The run queue length is an indicator of whether the system has sufficient CPU resources for all the processes it executes.

High values, along with high CPU utilization, indicate that the system requires faster or additional CPU units to handle the given load.

Samples

Number of Collector samples that took place during the specified time range.

Service Broker Wait

Time spent by the various sessions waiting for Service Broker event handlers and endpoints.

Session Logons

Number of sessions whose activity started during the current interval.

Session Logoffs

Number of sessions that logged out during the current interval.

SQL Agent CPU Usage

The CPU utilization percentage by the SQL Agent service.

SQL Agent Resident Memory Usage

The percentage of physical RAM (resident) memory consumed by the SQL Agent service.

SQL Executions

The number of SQL statements executed during the specified time range.

SQL Mail CPU Usage

The percentage of CPU used by the SQL Mail service.

SQL Mail Resident Memory Usage

The percentage of physical RAM (resident) memory used by the SQL Mail service.

SQL Recompilations

The number of SQL recompilations. A recompile occurs when an execution plan for an SQL statement needs to be changed, due to implicit or explicit decision.

Implicit recompilations occurs when the data structure (views, tables and their indexes) undergoes changes, or when their statistic data is updated. Explicit recompilations result from procedures that include explicit demand for recompilation (either in the procedure definition or in its execution), or from use of dynamic code, when parameterization is not forced.

Unusually high rate of recompilations vs. compilations ratio indicates that the re-compilation is inefficient, and prevents the use of existing, cached execution plans.

SQL Response Time

Average duration of the SQL statements that executed during the specified time range.

SQL Server Background CPU Usage

The percentage of CPU utilization by the SQL Server background services.

SQL Server Cache Memory

Total amount of dynamic memory the server is using for the dynamic SQL cache.

SQL Server Connections Memory

Total amount of dynamic memory the server is using for maintaining connections.

SQL Server Connections Summary

The total number of SQL Server sessions; both user and system sessions.

SQL Server Foreground CPU Usage

The percentage of CPU utilization by the SQL Server background services.

SQL Server Resident Memory Usage

The percentage of physical RAM (resident) memory utilized by SQL Server.

SQL Server Swap Memory Usage

The amount of virtual memory that the SQL Server process has reserved for use in the paging file (s).

Synchronous Task Wait

Time spent by the various sessions waiting for tasks started synchronously (most SQL Server processes are started asynchronously).

Table Lock Escalation

Lock escalation takes place when a specific query, which obtains a large number of locks at the row or page level, is automatically upgraded to a table lock level. This upgrade is being carried out as granting a single lock at this level; this is deemed more effective than creating and granting a many locks at lower levels.

Target Instance Memory

Total amount of dynamic memory the instance can consume.

Total CPU Usage

Overall CPU utilization by all processes running on the host (both OS and SQL Server processes).

Total Instance Memory

Total amount of dynamic memory the instance is currently consuming.

Virtual Memory Used

The current amount of used virtual memory.

Rules

Foglight for SQL Server includes an alarm template with a set of predefined rules that are applied to the incoming monitoring data. The rules that monitor SQL Server instances and databases have the prefix **DBSS**. You can and should modify alarm template rules to better suit your environment.

To modify Foglight for SQL Server rule threshold values and alarm template settings, refer to Customizing Alarms for Foglight for SQL Server Rules and “Viewing, Creating, and Managing Alarm Templates” in the *Foglight User Guide*.

Foglight Administration — Rules Management Dashboard

NOTE: Foglight 6.0 introduces alarm templates to gather alarm rules into a domain-specific template that is easily modified and applied to targets.

Avoid editing Foglight for SQL Server rules in the **Administration > Rules & Notifications > Rule Management** dashboard. Default rules may be modified during regular software updates and your edits will be lost. Always use the Alarm Templates dashboard.

In the Rules Management dashboard, all Foglight *for SQL Server* rules are displayed in the list. You can restrict the list to show only the rules installed with a cartridge. You can also search the list for a single rule or a group of rules.

To review rules:

1. In the navigation panel, under **Homes**, click **Administration**.
2. In the Administration dashboard, click **Rules**.

   The Rule Management dashboard opens.
3. To show only Foglight for SQL Server rules, in the Search box, type the prefix: **DBSS**.

   TIP: Alternatively, you can select DB_SQL_Server_UI from the Cartridge list.
4. From here, you can perform the following tasks:
   - Review a short description of the rule.
   - Associate actions with rules.

   NOTE: By default, predefined Foglight for SQL Server rules do not trigger any actions.

   - Create user-defined rules.

   NOTE: User-defined rules are always managed from the Rule Management dashboard; these rules are *not* displayed in the Foglight Administration — Rules Management Dashboard.

   For help with these tasks, open the online help from the Rules Management dashboard.

Collections and Metrics

The Foglight for SQL Server’s agent collects and stores data at all times, even when the browser window that displays the data is not active. The Collection Frequencies view in the Databases Administration dashboard includes a number of predefined collections that are sampled and stored, as well as the data collection values when sampling is carried out in online, offline, and real-time frequency modes.

For more information about working with collections, see the *Foglight Administration and Configuration Guide*.

NOTE: All of the information collected according to the definitions described in this section is displayed on Foglight for SQL Server. For details, see the Foglight for SQL Server User Guide.

NOTE: Several collections contain metrics whose name ends with Rate (for example: Physical Writes Rate and Physical Disks Rate). Such metrics, which are not documented in this guide, are used for plotting the original metric (usually called by the same name without the “Rate” suffix) on a graph for the selected time range.

SQL Server Agent’s Default Collections

Column	Description
Collection Name	The collection name, as retrieved by the agent.
Display Name	The collection name, as displayed in the application’s sections to which the collection retrieves information.
Description	The name of the collection. This list is sorted in alphabetical order.
Offline Frequency	Allows defining the collection interval, in seconds, in low frequency mode (for example: 300). Low frequency mode refers to the frequency at which the agent collects data when no user is currently connected to the monitored instance.
Online Frequency	Allows defining the collection interval, in seconds, in high frequency mode (for example: 60). High frequency mode refers to the frequency at which the agent collects data when none of the currently logged-on users are in focus on a screen, which displays any of the metrics sampled by this collection.
Real-time Frequency	Allows defining the collection interval, in seconds, in real-time frequency mode (for example: 20). Fast frequency mode refers to the frequency at which the agent collects data when one or more of the currently logged-on users are in focus on a screen, which displays any of the metrics sampled by this collection.

Access Methods

Purpose

This collection provides the methods used for accessing and updating logical data within an instance.

Type

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Forwarded Records	The total number of records that were retrieved by SQL Server during the specified time range. When a row in a table that *does not* have a clustering index is updated, and if the modified row no longer fits on the old page, SQL Server moves the row to a new page and leaves a forwarding pointer in the old page. As a result, the non-clustered indexes do not have to be modified to reflect the new row location, but will cause subsequent retrieval of this row, by means of non-clustered indexes to require an extra I/O. A high Forwarded Records rate can indicate a need for reorganizing tables (unloading/reloading) or defining clustered indexes.
Free Space Scans	The number of scans initiated to search for free space to insert a new record.
Full Scans	The number of unrestricted full scans per second (either base-table or full-index scans).
Index Searches	The number of index searches per second. These searches are used to start a range scan, reposition a range scan, revalidate a scan point, fetch a single index record, and search down the index to locate where to insert a new row.
Page Allocations	The total number of pages that were allocated to tables or indexes during the specified time range. This metric indicates how fast tables are expanding.
Page Allocations Rate	The rate per second at which pages are allocated to tables or indexes.
Page Deallocations	The total number of pages that were deallocated from tables or indexes during the specified time range. This metric indicates how fast tables are shrinking.
Page Deallocations Rate	The rate per second at which pages are deallocated from tables or indexes.
Page Splits	The number of page splits that took place within the defined time range. Page splitting occurs when an index or data page becomes full and then is split between the existing page and a newly allocated page. Excess page splitting can lead to I/O performance issues. High and increasing values may indicate that either the indexes needs to be rebuilt and / or the indexes fill factor should be increased.
Page Splits vs Page Writes	The ratio of page splits per writes. When an index or data page becomes full during an insert operation, data is split between the current page and a newly allocated page. Excessive page splitting can result in excessive disk I/O, thereby contributing to performance degradation. Higher values of this metric indicate that page splitting affects the monitored application's I/O performance. To resolve this issue, check your database file configuration, the selected index fill factor, and the rate at which your transactions are committed.
Probe Scans	The number of probe scans. A probe scan is used to directly look up rows in an index or base table.
Range Scans	The number of qualified range scans through indexes.

Display Name	Description
Skipped Ghosted Records	The total number of ghosted records that were encountered by SQL Server during scans over the specified time range. When a record is deleted from a table, SQL Server improves concurrency by not physically removing the row from index leaf levels, but marking it instead as deleted (ghosted). At some later point, a housekeeping process asynchronously removes these rows from the leaf level. Until the records are removed, SQL Server must skip the ghosted records during leaf-level scans.
Skipped Ghosted Records Rate	The rate per second at which SQL Server encounters ghosted records during scans.
Table Lock Escalations	The number of times a lock on a table has been escalated during the specified time range. Lock escalation is not necessarily a problem by itself; however, it can cause concurrency issues and lock conflicts, and can be a major contributor to blocking problems and deadlocks.

Agent Alert List

Purpose

This collection provides a list of all SQL Server Agent's alerts, including the type of event that triggered the alert (performance or alert event) and the number of occurrences for each alert.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	3600

Collection Metrics

Display Name	Description
Alert Name	The name of the alert that was raised.
Alert Type	The type of alert.
Database	The name of the database where the alert was raised.
Definition	A brief description of the specified alert.
Enabled	The status of the alert.
Last Occurred	The previous time the alert was raised.
Occurrences	The number of the alert’s occurrences.

Agent Job List

Purpose

This collection provides a list of all SQL Server Agent's scheduled jobs, including their status and duration.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	3600

Collection Metrics

Display Name	Description
Category	The SQL Agent job category assigned to this job.
Current Server Time	The Server time at the time the query ran.
Current Status	The current status of the job (Running or Inactive).
Current Step Number	Displays the current step number if the job is currently running.
Description	A brief description of the job.
Is Enabled	Determines whether the job can be run.
Job Name	The name of the job.
Job State	The current state of the job.
Jobs Failed Threshold	A flag indicating that a job has failed.
Jobs Retry Threshold	A flag indicating that a job retry has occurred.
Last Run Duration	Indicates how long the last run took (displayed in the d hh:mm:ss format).
Last Run Finish Time	The time the job finished.
Last Run Message	The last message associated with that step in the job.
Last Run Outcome	The result of the last run (Success or Fail).
Last Run Outcome Threshold	A flag indicating that the last run’s outcome is a job retry.
Last Run Time	The date/time on which the job last ran.
Long Running Job Threshold	A flag indicating that a long running job has occurred.
Next Run Time	The date and time of the next scheduled run.
Raise Alarm	Indicates whether the job raised an alarm the last time it ran.
Short Job Name	The short name of the job (in cases where the job name is too long to be displayed in full).

Always On

It includes three separated collections: Always On Cluster General, Always On Cluster Members, and Always On Cluster Network.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Cluster General

Collecting SQL Server Always On Cluster Configuration.

Collection Metrics

Display Name	Description
Cluster Name	Name of the WSFC cluster that hosts the instances of SQL Server that are enabled for AlwaysOn Availability Groups.
Quorum Type	Description of quorum_type, one of: NODE_MAJORITY, NODE_AND_DISK_MAJORITY, NODE_AND_FILE_SHARE_MAJORITY,NO_MAJORITY:_DISK_ONLY
Quorum State	State of the WSFC quorum, one of: 0 = Unknown quorum state 1 = Normal quorum 2 = Forced quorum
Member Name	Member name, which can be a computer name, a drive letter, or a file share path.
Member Type	The type of member, one of: 0 = WSFC node 1 = Disk witness 2 = File share witness

Cluster Members

Collecting SQL Server Cluster Members Configuration.

Collection Metrics

Display Name	Description
Member Name	Member name, which can be a computer name, a drive letter, or a file share path.
Member Type	The type of member, one of: 0 = WSFC node 1 = Disk witness 2 = File share witness
Member State	The member state, one of: 0 = Offline, 1 = Online
Network Subnet IP	Network IP address of the subnet to which the computer belongs. This can be an IPv4 or IPv6 address.

Cluster Network

Collecting SQL ServerCluster Network Configuration.

Collection Metrics

Display Name	Description
Member Name	A computer name of a node in the WSFC cluster.
Network Subnet IP	Network IP address of the subnet to which the computer belongs. This can be an IPv4 or IPv6 address
Network Subnet IPv4 Mask	The network subnet mask that specifies the subnet to which the IP address belongs. NULL = IPv6 subnet.
Network Subnet Prefix Length	Network IP prefix length that specifies the subnet to which the computer belongs.
Is Public	Whether the network is private or public on the WSFC cluster, one of: 0 = Private 1 = Public
Is IPv4	Type of the subnet, one of: 1 = IPv4 0 = IPv6

Availability Groups

It collects SQL Server Availability Groups.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Availability Group Name	Name of the availability group. This is a user-specified name that must be unique within the Windows Server Failover Cluster (WSFC) cluster.
Availability Group ID	Unique identifier (GUID) of the availability group.
Ag Resource ID	Unique ID of the availability group as a resource in the WSFC cluster.
Ag Databases Group ID	Unique WSFC databases Group ID of the availability group.
Primary Replica	Name of the server instance that is hosting the current primary replica. NULL = Not the primary replica, or, unable to communicate with the WSFC failover cluster.
Primary Recovery Health	Indicates the recovery health of the primary replica, one of: 0 = In progress 1 = Online NULL
Secondary Recovery Health	Indicates the recovery health of a secondary replica, one of: 0 = In progress 1 = Online NULL
Synchronization Health	One of: 0 = Not healthy 1 = Partially healthy 2 = Healthy This metric summarizes the synchronization status of all replicas in the availability group: • Healthy if all replicas are fully synchronized. • Partially healthy if only some of the replicas are fully synchronized. • Not healthy if no replica is fully synchronized.
Failure Condition Level	0 - No automatic failover or restart; 1 - Failover or restart on server down; 2 - Failover or restart on server unresponsive; 3 - Failover or restart on critical server errors; 4 - Failover or restart on moderate server errors; 5 - Failover or restart on any qualified failure conditions.

Display Name	Description
Health Check Timeout	Availability Group health check timeout in milliseconds.
Automated Backup Preference	At an availability group level, specify whether backups should or should not run on the primary replica. The possible preferences for where backups should run are as follows: 0 - PRIMARY - Only on the primary replica - Backups should always occur on the primary replica. 1 - SECONDARY_ONLY - Performing backups on a secondary replica is preferable. 2 - SECONDARY - Performing backups on a secondary replica preferable, but performing backups on the primary replica is acceptable if no secondary replica is available for backup operations. This is the default behavior. 3 - NONE - No preference about whether backups are performed on the primary replica or on a secondary replica.

Backup Locations

The collection shows drives or shares for backup locations.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	3600
Online	3600
Offline	3600

Collection Metrics

Display Name	Description
Member Name	The total free space of the backup media.
Error Message	It indicates that there has been an error, which trying to reach media for inspection.
Media Used Space	The total used space on the media.
Media Free Percentage	The percentage of free space out of the total size of the media.
Media Used Percentage	The percentage of the used space our of the total size of the media.

Blocking History

This collection retrieves the hierarchical tree of all sessions involved in the blocking process. The hierarchical tree is based on snapshots taken at a predefined interval.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	60
Offline	60

Collection Metrics

Display Name	Description
Blocked	The total number of all blocked sessions.
Blocked by	The session that is currently blocking the selected session.
Blocked Time	Total wait time spent by all blocked sessions.
Command	The current or previous command executed.
CPU	The total amount of CPU consumed so far by the session. This metric can be useful when deciding which sessions to kill.
Database	The name of the database.
Host Name	The name of the client computer.
Lock Type	The type of the resource that is currently locked (Database, Table, Page, Row, Extent, and so on).
Locking	The unique session number for the session that either owns or requests the lock.
Login Time	The time that the session was created.
Mode	The kind of lock being applied to the resource (Shared, Exclusive, Update, IntentShared, IntenExclusive and so on).
OS User	Windows user name for a session that uses Windows Authentication.
Physical IO	The total amount of I/O operations carried out so far by the session. This metric can be useful when deciding which sessions to kill.
Program	The name of the application that the user is currently running (for example, Microsoft Access).
Request Status	• The status of the lock; can have one of the following values: • WAIT = Wait for another resource. • CNVT = Converted to another resource. • GRANT = the lock has granted.
Resource	The resource that is in conflict. This metric often identifies a database and table. The data in the Resource column is reported directly from SQL Server and, for performance reasons, is not resolved to actual resource names during normal data collection.
SQL Text	Displays the SQL text of the session that is blocked and/or blocking.
SQL User	The SQL Server user login name for this session.
Status	The status of the session (Blocked, Blocking or both). For sessions at the head of the blocking chain (those that are not blocked), this metric also indicates if the session is Runnable or Sleeping.
Type	The type of the lock request that is waiting (for example: Database, Table, Page, Row, Key, and Extent).
Wait Time	Indicates how long this session has been waiting for the lock (measured in seconds). This shows 0 if the session is not waiting.

Blocking List

Purpose

This collection provides a display of a hierarchical list of sessions, which sorts the blocked and blocking sessions in a logical order.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Blocked By SPID	Indicates which SPID (if any) holds locks on a resource on which this session is waiting.
Command	The current or previous command executed. This information can be useful when deciding which sessions to kill.
CPU	The total amount of CPU consumed by the session so far. This information can be useful when deciding which sessions to kill.
Database	The name of the database.
Host Name	The name of the Client computer.
OS User	Windows user name for a session that uses Windows Authentication.
Physical I/O	The total amount of I/O consumed by the session so far. This information can be useful when deciding which sessions to kill.
Program	The application program that the user is using (for example, Microsoft Access).
Resource	The resource that is in conflict. This value often identifies a database and table. The Resource data is reported directly from SQL Server.
SPID	The session unique identifier (Session Process ID).
SQL Text	Displays the SQL belonging to the session that is blocked and/or blocking.
SQL User	The SQL Server user login name for this session.
Status	The status of the session (Blocked, Blocking or both). For sessions at the head of the blocking chain (those that are not blocked), this also indicates if the session is Runnable or Sleeping.
Type	The type of the lock request that is waiting (Database, Table, Page, Row, Key, Extent and so on).
Wait Time	Indicates how long this session has been waiting for the lock (measured in seconds). If the value displayed is 0, the session is not waiting.

Buffer Cache List

Purpose

This collection provides a detailed list of buffer cache objects.

Type

SQL Server.

Collection Sampling Intervals

Collection Metrics

Frequency Mode	Collection Interval (In Seconds)
Real-time	60

Display Name	Description
% Object Dirty	Percentage of object modified pages.
% Of Cache	The percentage of cache used by the object.
% Of Object	The percentage of object in the cache.
Cached MB	The amount of cache used by the object.
Database	The name of the database where the object resides.
Dirty Size	Size of object modified pages in megabytes.
File Group	The file group where the object resides.
Index	The index name (if the object is an index). If this column is left empty, the object is a heap.
Index ID	The index identifier (if the object is an index). This metric can have one of the following values: • 0 — Heap • 1 — Clustered index • > 1 — Non-clustered index
Object Size	Object allocation size in megabytes.
Owner	The user name of the object's owner.
Row ID	The row ID provides a unique identification of each row.
Table	The table name (if the object is a table).
Unused Allocated Cache	The amount of allocated cache that is not used by the object.

Buffer Manager

Purpose

This collection manages SQL Server's use of data within the logical data layer and monitors the server's physical I/O operations.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Buffer Cache Free Size	The total number of free pages on all free lists. This value is calculated as follows: <free pages> * 8K / 1024. When SQL Server needs a free memory page (for example, when reading a database page from disk into the buffer cache), and if no free pages are immediately available, the connection needing the free page will have to wait while SQL Server makes buffers available. This will result in slower performance. In the worst case, the connection will have to wait while SQL Server writes a modified page out to disk in order to make a free buffer.
Buffer Cache Hit Rate	The total number of cache hits.
Buffer Cache Hit Ratio	The ratio of physical reads to logical reads. It indicates the percentage of database page I/O requests that were satisfied from the Buffer Cache and therefore did not have to perform disk reads. This ratio measures how efficiently SQL Server is using the memory allocated to its buffer cache.
Buffer Cache Reads	The total number of cache lookups over the last few thousand page accesses.
Buffer Cache Size	The amount of memory currently allocated to the buffer cache, including database, free, and stolen pages. This size is calculated as follows: <free pages> * 8K / 1024. The Buffer Cache is an in-memory copy of recently used database pages.
Checkpoint Pages	The number of pages flushed by checkpoint or other operations that require all dirty pages to be flushed.
Database Pages	The number of pages that constitute the SQL data cache. A large change in this value indicates the database is swapping cache values from the cache. This database pages value is calculated as follows: <pages> * 8KB / 1024. SQL Server keeps a copy of its most recently used database pages in the buffer cache. When a connection needs to reference a database page, SQL Server performs a Logical I/O operation by checking the buffer cache to see if the requested page is already in memory. If the page is found in the buffer cache, a logical I/O read will be carried out; otherwise, the page is read from disk, using a Physical I/O operation.

Display Name	Description
Free Pages	The total number of free pages on all free lists. This value is calculated as follows: <free pages> * 8K / 1024. The Lazy Writer Process periodically scans all SQL Server caches, and maintains a list of such “free” pages. When SQL Server needs a free memory page (for example, when reading a database page from disk into the buffer cache), and no free pages are immediately available, the connection needing the free page must wait while SQL Server makes buffers available. This results in slower performance. In the worst case, the connection will have to wait while SQL Server writes a modified page out to disk in order to make a free buffer.
Lazy Writes	The number of buffer pages that have been modified in the buffer cache, which the Lazy Writer process is flushing to disk. A high value of lazy writes may indicate that SQL Server is running out of available space in the buffer pool cache.
Logical Page Reads	The number of logical database page reads issued. Under optimal work conditions, the SQL Server connections use logical page reads, that is: reference pages from the buffer cache. However, if the required page is not yet in the cache, it is read from disk using physical I/O operations. A high value of the logical page reads indicates that SQL Server efficiently uses the memory allocated to its buffer cache. A high value of physical page reads, on the other hand, indicates that SQL Server is finding fewer pages already in memory, and therefore has to perform more disk reads.
Page Life Expectancy	The value of page life expectancy, that is: the length of time in seconds that a database page will stay in the buffer cache without being accessed, before it is flushed out. Microsoft recommends keeping this value greater than five minutes (300 seconds). Values smaller than 300 indicate that pages are being flushed out of the cache within a small period of time. The resulting lack of pages in the buffer cache requires SQL Server to carry out more disk reads, thereby degrading its performance.
Physical Page Reads	The number of physical database page reads issued. Physical page reads are used when a connection requests a page that is not already in the buffer cache.
Physical Page Writes	The number of physical database page writes issued. Normally, SQL users do not have to wait for database write operations to complete. Most modifications to database pages are made in the buffer cache.

Display Name	Description
Read Ahead Pages	The number of pages being read from disk before they are requested, using the Read Ahead process. Read ahead is the process by which SQL Server anticipates which pages a session will require, and reads them into the Buffer Cache before they are requested by the user connection. This process is used when SQL Server determines that a session is accessing table or index pages in a sequential manner. If the sequential processing does continue, these pages will be in the cache memory when required, thereby avoiding the need to wait for an I/O operation. Read Ahead can improve queries' performance when accessing a new table or an existing table with a new filter. However, a high level of Read Ahead activity may indicate poorly coded SQL statements or inadequate indexes.
Read Ahead Pages Rate	The rate per second at which pages were read from disk before being requested, using the Read Ahead process.
Reserved Pages	The number of buffer pool reserved pages. The reserved pages value is calculated as follows: <pages> * 8KB / 1024.
SQL Server Physical I/O Operations	Total number of pages read or written to disk. Even though some of these operations may be satisfied by the file system cache, they usually require physical disk access.
Stolen Pages	Pages "stolen" from the buffer pool to be used for miscellaneous server purposes, when Windows requires memory for another application. A large number of stolen pages may be normal for a short period (for example: if the system backup begins after a large database batch run completes). However, a constantly high value of this metric may indicate overall system memory shortage.
Target Pages	The ideal number of pages in the buffer pool. The target pages value is calculated by <pages> * 8KB / 1024.

CLR Assemblies

Purpose

This collection provides information on the security state of CLR (Common Language Runtime) assemblies.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Database	The database name
Is Hidden	Indicates whether there is a hidden CLR assemble within the selected database.
Is External	Indicates whether there is an external CLR assemble within the selected database.
Is Unsafe	Indicates whether there is an unsafe CLR assemble within the selected database.

Cluster Summary

Purpose

This collection provides information on the state of the currently monitored cluster environment, as shown in the Cluster Administration application.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
ID	The unique identifier for the specified resource in the cluster.
Parent ID	The parent ID of the node’s vertical level.
Type	The type of resource (Resource group).
Name	The resource name.
Status	The status of the cluster resource.
Comment	A brief description of the specified cluster resource (if available).
Other Data	The group to which the resource belongs.
Icon Flag	A flag indicating whether there is an error.
Take Offline Cmd	A flag indicating the existence of a command that runs when the resource goes offline.
Bring Online Cmd	A flag indicating the existence of a command that runs when the resource goes online.
Move Group Cmd	A flag indicating the existence of a command that runs when the resource is moved to another group.
Stop MSCS Cmd	A flag indicating the existence of a command that runs when the resource is stopped.

Display Name	Description
Start MSCS Cmd	A flag indicating the existence of a command that runs when the resource is started.
Running On Preferred Server	A flag indicating whether the node is running on a server that appears on the preferences list.
Controls The Current SQL Server	A flag indicating whether the node is currently controlling the SQL Server.
Running On Server	Indicates the host (server) on which the selected node is running.
Non Active Node Available	Indicates which non-active SQL Server nodes are currently not running.
Can Accept Move	A flag indicating whether the resource can accept being moved to another group.

Configuration

This collection provides a list of the values that are configured and running in the SQL Server instance.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	3600
Online	3600
Offline	86400

Collection Metrics

Display Name	Description
Config Value	The value to which the configuration option is set. This value may not be implemented immediately upon creation. Usually, restarting SQL Server is necessary before any change takes effect.
Description	A description of the specified configuration option.
Last Change	The date when the value was last changed.
Last Run Value	The most recent value, set during the last modification.
Max	The maximum permitted value for the configuration option.
Min	The minimum permitted value for the configuration option.
Modifiable	Indicates when changes to the option take effect. The following options are possible: • Immediate: changes to these options take effect immediately. • Restart: changes to these options take effect after SQL Server is stopped and restarted. • Never: these options cannot be modified.

Display Name	Description
Name	The name of the specified configuration option.
Run Value	The value currently used by the configuration option.

Database Index Density Vectors

This collection provides a list of density values for each combination of columns in a specific database index.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
All Density	Density is the term used by SQL Server to represent the selectivity of the index columns. The more selective an index is, the more useful it is in searches. Calculated as 1 / distinct values for all values in the first key column of the statistics object, excluding the histogram boundary values. This Density value is not used by the query optimizer, and is displayed for backward compatibility with versions prior to SQL Server 2008.
Avg Key Length	Average number of bytes per value for all of the key columns in the statistics object.
Columns	The names of the columns in the index.

Database Index Details

This collection provides general statistics about a specific database index.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Density	Density is the term used by SQL Server to represent the selectivity of the index columns. The more selective an index is, the more useful it is in searches. Calculated as 1 / distinct values for all values in the first key column of the statistics object, excluding the histogram boundary values. This Density value is not used by the query optimizer, and is displayed for backward compatibility with versions prior to SQL Server 2008.
Last Updated	The time and date when the statistics were last updated.
Rows	The number of rows in the index.
Rows Sampled	Number of rows sampled for statistics information.
Steps	The number of distribution steps.

Database Index Fragmentation Info

This collection provides information about index fragmentation in the currently monitored database.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Database	The database ID.
Index	The index ID.
Index Property	The property name of each element of the index.
Owner	The schema owner.
Property Value	The value of each element of the index.
Table	The table ID.

Database Index Histogram

This collection provides the distribution of the range key and values within a specific database index.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Avg Range Rows	Average number of rows that corresponds to the last high range and the current one inclusive (Range Rows / Distinct Range Rows for Distinct Range Rows > 0).
Distinct Range Rows	Number of distinct rows that correspond to the last high range and the current one inclusive.
Equal Rows	Number of rows that correspond to the current high range.
Range High Key	The upper bound value of a histogram step.
Range Rows	Number of rows that corresponds to the last high range and the current one inclusive.

Database Index List

This collection provides the distribution of the range key and values within a specific database index.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
DB Name	The name of the specified database.
FG Name	The name of the File Group where the index resides.
Free MB	The free size left for the index to grow, in megabytes.
Index Id	Unique identifier of the index.
Index MB	The size of the index at the time of its last statistics update, measured in megabytes.
Index Name	The name of the index.
Num Keys	The number of keys held by the index.
Orig Fill Factor	Represents the size the index occupies as a percentage of the total index page (8 KB).
Owner Name	The owner of the index.
Qualified Name	The full qualified name of the database object.
Row Mod Ctr	The number of row modifications made since statistics were last updated for this index.
Rows	The number of rows in the table.
Table Name	The table with which the index is associated.
Type	The index type.
Used MB	The actual size occupied by the index, in megabytes.

Database Information

Purpose

This collection monitors the database's operations, statistics and other general data.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	3600

Collection Metrics

Display Name	Description
% Log Used	The percentage of used space within the file space allocated to the transaction log of each currently used database. Log space can be freed up by backing up the log, or truncating it, using the truncate option (`backup log <dbname> with truncate_only`). All log records that exist after the oldest open transaction cannot be freed.
Average Active Transactions	Average number of active transactions during the specified time range.
Average Log Flush Wait Time	Average amount of time (in milliseconds) spent waiting for log flushes in each database. A high log flush wait time can result from a slow or overworked disk subsystem. If a database has a consistently high value of log flush wait time, with no changes, run the SQL command CHECKPOINT on that database to force another log flush and check the value again.
Average Transactions	Average transactions for each database.
Average Transactions Rate	The average rate per second of transactions executed for each database.
Backup Throughput MB	Data transition size (in MB) of backup or restore commands that are reading or writing to the database. While backups are essential, other users will probably experience performance degradation while these operations run.
Backup Throughput Rate	The rate (measured in megabytes per second) at which backup or restore commands are reading or writing to the database. While backups are essential, other users may experience performance degradation while these operations run.
Bulk Copy Throughput	Data transition size (in MB) of data being loaded into the database using BCP (Bulk Copy Program) or BULK INSERT. While BCP and BULK INSERT are the fastest way of getting data into SQL Server, other users will in most probability experience performance degradation while these operations run.
Bulk Copy Throughput Rate	The rate (measured in megabytes per second) at which data is being loaded into the database using BCP (Bulk Copy Program) or BULK INSERT. Despite the top speed at which BCP and BULK INSERT can import data into SQL Server, system I/O performance may degrade while BCP operations are underway.
Database ID	The database ID.
Database Name	The name of the database.
DBCC Scans	The size (in MB) of data transition resulting from data logical read scans carried out by Database Console Commands (DBCC).
Log Cache	The total number of reads that were satisfied from the log cache, thereby avoiding physical log I/O operations.
Log Cache Hit Rate	The rate per second of log cache reads, namely: reads that were satisfied from the log cache, thereby avoiding physical log I/O operations.

Display Name	Description
Log Cache Hit Ratio	The percentage of reads that were satisfied from the log, thereby avoiding physical log I/O operations.
Log File Size	The amount of disk space that the log files are using.
Log Flush Wait Time	The amount of time spent waiting for log flushes in each database. The log flush operation, which is necessary to guarantee that transactions can be recovered in the event of a system failure, is most commonly caused when a user is issuing a commit transaction. A high log flush wait time can result from a slow or overworked disk subsystem. If a database has a consistently high value of log flush wait time, run the SQL command CHECKPOINT on that database to force another log flush, and then recheck the value.
Log Flushes	The total number of log pages that were being written to disk by the log writer process during the specified time range.
Log Flushes Rate Total	Number of times per second that the log caches of all of the instance's databases are being flushed to disk.
Log Growth Count	The number of times the log has been expanded for each database.
Log Growth Count Rate	The number of times per second the log has been expanded for each database.
Log Reads	Log cache lookups.
Log Shrink Count	The number of times the log has been reduced for each database.
Log Truncation Count	The total number of log truncations performed for each database during the specified time range. Log truncations can be carried out using either of the following methods: • Manual truncations (backup log <dbname> with truncate_only) • Automatic truncations — in either of the following settings: ▪ The database has the *trunc.log on chkpt.* option turned on ▪ The recovery model is set to Simple
Total Data Size	The amount of disk space that the data files are using.

Database Properties

This collection displays a list of databases and their overall properties, such as single user, status and used collation.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	3600
Online	3600
Offline	86400

Collection Metrics

Display Name	Description
Ansi Null Default	Database follows SQL-92 rules for allowing null values.
Ansi Nulls	All comparisons to a null evaluate to unknown.
Ansi Padding	Strings are padded to the same length before comparison or insert.
Ansi Warnings	Error or warning messages are issued when standard error conditions occur.
Arithmetic Abort	Queries are ended when an overflow or divide-by-zero error occurs during query execution.
Auto Close	Database shuts down cleanly and frees resources after the last user exits.
Auto Create Statistics	Existing statistics are automatically updated when the statistics become out-of-date because the data in the tables has changed.
Auto Shrink	Database files are candidates for automatic periodic shrinking.
Auto Update Statistics	AUTO_UPDATE_STATISTICS database option is enabled.
Bulk Copy	Database allows non-logged operations.
Close Cursors On Commit	Cursors that are open when a transaction is committed are closed.
Collation	Default collation name for the database.
DB Name	The database’s name provides a unique identification of the database within a SQL Server instance.
DBID	The database’s ID provides a unique identification of the database within a SQL Server instance.
Full Text	Database is full-text enabled.
Local Cursors Default	Cursor declarations default to LOCAL.
Merge Published	The tables of a database can be published for merge replication, if replication is installed.
Null Concat	Null concatenation operand yields NULL.
Numeric Round Abort	Errors are generated when loss of precision occurs in expressions.
Quoted Identifiers	Double quotation marks can be used on identifiers.
Recovery	Recovery model for the database.
Recursive Triggers	Recursive firing of triggers is enabled.
Status	Database status.
Subscribed	Database is subscribed to a publication.

Display Name	Description
Page Verify	The SQL Server Database Engine detects incomplete I/O operations, resulting from power failures or other system outages. This metric indicates the method used by the database for detecting incomplete I/O operations, and has the following options: • None • Checksum — A sum derived from the bits of a segment of computer data, which is calculated before and after transmission or storage. • Torn page detection — detection of an incomplete I/O operation
Trunc Log	Database truncates its logon checkpoints.
Updateability	Indicates whether data can be modified.
User Access	Indicates which users can access the database.

Database Sessions (Session List)

Provides a detailed list of the SQL Server sessions.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	300

Collection Metrics

Display Name	Description
Status	The session Status (runnable, sleeping, blocked, etc).
Last Command	The current or previous command executed.
Lock Timeout	The amount of time that this session will wait for lock requests to be satisfied. Corresponds to the @@Lock_Timeout global variable for this session.

Database Summary

This collection provides general information about the currently monitored SQL Server databases, such as availability, number of files and file groups, and capacity use.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	3600

Collection Metrics

Display Name	Description
Available Databases	The number of available databases.
Data Files	Number of data files.
Data Files Size	Total size of the data files.
Data Files Space	Indicates the total size of the data files used and the data files maximum size in percent.
File Groups	Number of file groups.
Log Files	Number of log files.
Log Files Size	Total size of the log files.
Log Files Space	Indicates the total size of the log files used size and the log files maximum size in percent.
Unavailable Databases	The number of unavailable databases.

Database Tables List

This collection provides a list of all of the database tables, as well as their capacity usage.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
DB Name	The name of the database that contains the specified table.
FG Name	The file group where the table is stored.
Free MB	The amount of free space in the table.
Owner Name	The name of the table owner.
Percent Of DB	The percentage of space that the table occupies in the database.
Qualified Name	The full qualified name of the database object.
Reserved MB	The amount of space reserved for the table in megabytes.
Row ID	The row identifier.
Rows	The number of rows in the table.

Display Name	Description
Table MB	The actual size of the table in megabytes.
Table Name	The name of the table.
Used MB	The amount of used space in the table.

Databases

Purpose

This collection displays a list of the databases and their status.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	14400

Collection Metrics

Display Name	Description
# Index	The number of indexes that exist on the database.
# Table	The number of tables, where table type=user, which exist on the database.
Database	The name of the database, unique within an instance of SQL Server.
Database Unavailable Threshold	A threshold which, when exceeded, causes an alarm to fire regarding a specific database that is unavailable.
Days Since Last Backup	The number of days that passed since the last database backup.
Days Since Last Backup for Rule	A threshold which, when exceeded, causes an alarm to fire regarding the number of days that have elapsed since the last database backup.
Days Since Last Backup no Backup	A threshold which, when exceeded, causes an alarm to fire regarding a specific database that has never been backed up.
DBID	ID of the database, unique within an instance of SQL Server.
Last Backup	The date and time of the most recent backup of any type that has been carried out for the database.
Oldest Trans SPID	The system process ID of the session owning the oldest transaction in the database.

Display Name	Description
Oldest Trans Start Time	The date and time at which the oldest transaction in the database began.
Status	The status of the database can be one of the following: • Unavailable • Read Only • No Access • Normal.

Deadlock

Purpose

This collection displays information about a deadlock situation.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	300

Collection Metrics

Display Name	Description
Event Sequence	The event sequence number that identifies the deadlock occurrence.
Start Time	The time at which the deadlock event started.
Text Data	The XML data structure, holding the deadlock full data.

DTC Information

This collection retrieves information on the transaction management carried out by the DTC (Distributed Transaction Coordinator).

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Aborted Transactions	The number of distributed transactions that were rolled back.
Active Transactions	The number of currently active distributed transactions.
Committed Transactions Sec	The number of distributed transactions committed per second.
In Doubt Transactions	Transactions that have passed phase 1 of the two phase commit (have committed the local transaction), and are awaiting a response from the DTC to either commit or roll back (phase 2).
Response Time Average	The average of the full time (in milliseconds) it has taken a query (select 1, by default) to get from the application to SQL Server and back.
Transactions Sec	The number of distributed transactions performed per second.

Error Log

The Error Log collection retrieves all error messages that were added to the SQL Server error log during the specified time range.

IMPORTANT:

IMPORTANT: The Error Log Check collection retrieves information only for SQL Server 2005 and later.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	900

Collection Metrics

Display Name	Description
Errors	The number of errors detected within the SQL Server error log, during the period specified for the collection frequency on the **Databases > Administration > Collection Frequency** view.
Check Date	Indicates the last date when the error log check was carried out.

Error Log List

The Error Log List collection provides a list that indicates the SQL Server existing error log files, their order and sizes.

NOTE:

NOTE: The list displays the logs in the following order: <log_name>-<log_files>-<log_size>.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	3600

Collection Metrics

Display Name	Description
Archive No	The sequence number of the log creation, from low (current = 0) to high.
Created	The time when the file was created.
Log Files	The log file name and size.
Log Name	The name of the Archive log.
Log Size	The total size of the log, in megabytes.
Log Type	The log type: • SQL Server — in which case the log appears on the SQL Server Error Logs pane • SQL Agent — in which case the log appears on the SQL Server Error Logs pane

Error Log Scan

This collection provides a list of the log details of a predefined error log filter.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Log Date	The date when the error log message was created.
Message	The error log message.
Process Information ((in versions higher than 2000 - Error Number).	Displays the event to which the data refers (server process, SPID and so on).
Row ID	The row unique identifier.

File Groups

Purpose

This collection displays a list of all SQL Server instance file groups and transaction logs details.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
% Free	The percentage of space not allocated to any objects.
% Used	The percentage of space allocated to objects.
Can Grow	Indicates whether the file can grow.
Database	The database's name provides a unique identification of the database.
DBID	The database's ID provides a unique identification of the database.
File Count	The number of files in each file group.
File Group	The name of the file group to which the file belongs.
Free Size	The amount of unused space, measured in megabytes.
Size	The size of the file, in megabytes.

Display Name	Description
Type	The type of file group (for example: data or log).
Used Size	The size used by the database file (in megabytes).

File Data Flow Statistics

Purpose

This collection displays a list of data and log files within a SQL Server instance, which provides detailed statistics about usage, size, and data flow.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
Current Reads Wait Time	The number of seconds SQL Server has spent waiting for physical read operations on this file since the last time data was collected. Shows data only for SQL Server 2005.
Current Wait Time	The number of seconds SQL Server has spent waiting for I/O operations on this file since the last time data was collected.
Current Writes Wait Time	The number of seconds SQL Server has spent waiting for write operations on this file since the last time data was collected. Shows data only for SQL Server 2005.
DB And File Group	The database name, concatenated with its file group name.
DB File Name	The database name, concatenated with its file name.
DB Name	The name of the SQL Server database that contains the file.
Disk	The disk where the file resides.
File Group	File group name.
File Name	The name of the file.
MB On Disk	Physical file size on disk in MB. Shows data only for SQL Server 2005.
MB Read	The number of MB read from this file since SQL Server started.
MB Written	The number of MB written to this file since SQL Server started.
Path	The name of the file whose I/O statistics are on display.
Reads	Number of reads issued on the file.
Reads Rate	The current rate at which SQL Server is performing physical read operations from this file.

Display Name	Description
Reads Wait Time	The total number of seconds SQL Server has spent waiting for physical read operations on the file since SQL Server started. Shows data only for SQL Server 2005.
Total IO	The number of MB read from and written to this file since SQL Server started (MB Read plus MB Written).
Total IO Rate	The current rate at which SQL Server is performing both physical read and write operations for this file (Reads Rate plus Writes Rate).
Total Wait Time	The total number of seconds SQL Server has spent waiting for I/O operations on this file since SQL Server started.
Writes	Number of writes issued on the file.
Writes Rate	The current rate at which SQL Server is performing physical write operations to this file.
Writes Wait Time	The total number of seconds SQL Server has spent waiting for write operations on the file since SQL Server started. Shows data only for SQL Server 2005.

File Groups

Purpose

This collection displays a list of all SQL Server instance file groups and transaction logs details.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
% Free	The percentage of space not allocated to any objects.
% Used	The percentage of space allocated to objects.
Can Grow	Indicates whether the file can grow.
Database	The database's name provides a unique identification of the database.
DBID	The database's ID provides a unique identification of the database.
File Count	The number of files in each file group.
File Group	The name of the file group to which the file belongs.
Free Size	The amount of unused space, measured in megabytes.
Size	The size of the file, in megabytes.

Display Name	Description
Type	The type of file group (for example: data or log).
Used Size	The size used by the database file (in megabytes).

Files

This collection displays a detailed list of all data and log files within a SQL Server instance.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
% Free	The percentage of space not allocated to any objects.
% Used	The percentage of space allocated to objects.
Auto Grow	Whether the file can grow automatically as it fills.
Can Grow	Indicates whether any of the files in the transaction log can grow.
Data Files Used Percentage Threshold	A flag indicating a threshold for the used percentage of space within a data file. When this percentage exceeds the predefined threshold, an alarm will be invoked.
Database	The name of the specified database.
DB and File	The database name, concatenated with the database file's name.
DBID	The database’s ID; provides a unique identification of the database.
File	The name of the data file.
File Group	The name of the file group to which the file belongs.
Free Size	The amount of unused space, measured in megabytes.
Growth Increment	Indicates the amount by which the file will grow every time, if the autogrow option is enabled. For example: if the data file’s initial size is 1 MB and it has to grow to a final size of 11 MB, setting the Growth Inc. parameter to 2 MB causes the file to grow five times, each time by a 2 MB increment.
Growth Remaining	The number of times a log file can grow automatically before the file is almost full and can no longer automatically grow enough to relieve the problem.
Growths Remaining Data File Threshold	A flag indicating a threshold for the number of times the data file can AutoGrow until it is unable to grow anymore.
Growths Remaining Log File Threshold	A flag indicating a threshold for the number of times a log file can AutoGrow until it is unable to grow anymore.

Display Name	Description
Log File Used Percentage Threshold	A flag indicating a threshold for the used percentage of space within a log file.
Path	The path of the database file.
Size	The file's size.
Type	The type of file group. For example, data or log.
Used Size	The amount of space size used by the database file.

Files Drive Total

This collection, which is active only when monitoring SQL Server instances on a virtualized environment, provides a summary of all data and log files within a SQL Server instance, per disk drive.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	14400

Collection Metrics

Display Name	Description
Logical Disk	The logical drive name (for example: C, D).
Allocated Space	Total space (in megabytes) allocated to SQL Server files on the selected logical disk.
Used Space	Total space (in megabytes) used by SQL Server files on the selected logical disk.

Files Instance Summary

This collection provides a summary of all data and log files within a SQL Server instance.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900

Frequency Mode	Collection Interval (In Seconds)
Online	900
Offline	18000

Collection Metrics

Display Name	Description
Allocated Space	Total space (in megabytes) allocated to SQL Server files on all drives, including the unoccupied space.
Used Space	Total space (in megabytes) used by SQL Server files on all drives.

Full Text Catalog

This collection provides general details of full text indexes.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	14400

Collection Metrics

Display Name	Description
Database Name	The database's name provides a unique identification of the database within a SQL Server instance.
FT Catalog ID	The ID of the full-text catalog, which uniquely identifies it across the full-text catalogs in the database.
FTC Name	The full-text catalog's name provides a unique identification of the full-text catalog within a SQL Server instance.

Display Name	Description
FTC Status	The full-text catalog's status can be one of the following: • 0 = Idle • 1 = Full population in progress • 2 = Paused • 3 = Throttled • 4 = Recovering • 5 = Shutdown • 6 = Incremental population in progress • 7 = Building index • 8 = Disk is full. Paused. • 9 = Change tracking
Populate CompleteAge	The difference in seconds between the completion of the last full-text index population and 01/01/1990 00:00:00. This value is updated only for full and incremental crawls, and returns 0 if no population has occurred.
Table Count	The number of tables in the database that take part in the full text indexing process.
Item Count	The number of full-text indexed items currently stored in the full-text catalog.
Index Size MB	Size of the full-text catalog in megabytes.
Path	Name of the catalog directory in the file system.

InMemory OLTP (XTP)

The collection shows amount of InMemory OLTP (XTP) Memory usage.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	900

Collection Metrics

Display Name	Description
Database ID	Database ID
Database Name	Database name
Resource Pool Name	The name of the resource pool that is mapped to this database. This resource pool controls total memory available to memory-optimized tables in this database.
Total Allocated Tables	The total amount of memory allocated for the tables.

Display Name	Description
Total Allocated Indexes	The total amount of memory allocated for the Indices.
Total Allocated System	The total amount of memory allocated for the system.
Pool Consumption Percentage	Total database memory out of available pool size.
Total Used Tables	The total amount of memory used by the tables, including row versions.
Total Used Indexes	The total amount of memory used by the indices, including row versions.
Total Used System	The total amount of memory used by the system, including row versions.

Instance Wait Categories

Purpose

This collection provides data for a list of SQL processes' wait events, divided by resource event categories.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Active Time	Sum of all active wait events; equal to the session total activity within the specified time range.
Backup Recovery Wait	The average number of active sessions waiting for the completion of backup or recovery tasks during the selected time range.
CLR Wait	Time spent by the various sessions waiting for CLR code execution to complete.
CLR Wait Percent	Percentage of time spent by the various sessions waiting for CLR code execution to complete.
CPU Usage	Time spent by the various sessions consuming CPU cycles. This reading is taken directly from the operating system, rather than SQL Server wait states.
CPU Usage Percent	Percentage of time spent by the various sessions consuming CPU cycles. This reading is taken directly from the operating system, rather than from SQL Server wait states.
CPU Usage Rate	Percentage of time spent by the various sessions waiting for CLR code execution to complete.

Display Name	Description
CPU Wait	Time spent by the various sessions waiting in the systems run queue for CPU cycles. This reading is calculated from the operating system readings, rather than SQL Server wait states.
CPU Wait Percent	Time spent by the various sessions waiting in the system's run queue for CPU cycles. This reading is taken directly from the operating system, rather than from SQL Server wait states.
CPU Wait Rate	Time spent per second by the various sessions waiting in the system's run queue for CPU cycles. This reading is taken directly from the operating system, rather than from SQL Server wait states.
Cursor Synchronization Wait	Time spent by the various sessions waiting for cursor synchronization operations to complete.
Cursor Synchronization Wait Rate	The average number of active sessions waiting for the completion of cursor synchronization operations during the specified time range.
Database Replication Wait	Time spent by the various sessions waiting for replication synchronization events to complete.
Deferred Task Worker Wait	Time spent by the various sessions waiting for I/O requests; this wait indicates that either a large number of suspect pages are encountered or the disk subsystem is overloaded.
Distributed Transaction Wait	Time spent by the various sessions waiting for various distributed transaction events to complete.
External Procedure Wait	Time spent by the various sessions waiting for external procedures to end.
Full Text Search Wait	Time spent by the various sessions waiting for full text synchronization operations.
Hosted Components Wait	Time spent by the various sessions waiting for hosted components, such as, but not exclusively, CLR.
I/O Bulk Load Wait	Time spent by the various sessions waiting for a bulk load I/O to finish.
I/O Completion Wait	Time spent by the various sessions waiting for I/O operations to complete.
I/O Data Page Wait	Time spent by the various sessions waiting to latch a buffer for an I/O request; this wait is related to issues with the disk I/O subsystem.
I/O Wait	Time spent by the various sessions waiting for physical I/O operations to complete.
Idle	Idle time.
Internal Cache Latch Wait	Time spent by the various sessions waiting for latches that are not buffer latches or savepoint latches. Typically, this event can only be treated by partitioning data to provide more cache pages.
Latch Buffer Wait	Time spent by the various sessions waiting to latch a buffer that is not an I/O request.
Latch Savepoint Wait	Time spent by the various sessions waiting to synchronize commits to marked transactions.
Latch Wait	Time spent waiting for internal locks (latches) to be released.
Lock Bulk Update Wait	Time spent by the various sessions waiting to acquire bulk update locks.
Lock Exclusive Wait	Time spent by the various sessions waiting to acquire exclusive locks.
Lock Intent Wait	Time spent by the various sessions waiting to acquire intent locks.
Lock Schema Wait	Time spent by the various sessions waiting to acquire schema locks.

Display Name	Description
Lock Shared Wait	Time spent by the various sessions waiting to acquire shared locks.
Lock Update Wait	Time spent by the various sessions waiting to acquire update locks.
Lock Wait	Time spent by the various sessions waiting for a blocking lock (held by another session) to be released.
Log Buffer Wait	Time spent by the various sessions waiting for space in the log buffer or otherwise waiting for memory to be made available to write log records.
Log Other Wait	Time spent by the various sessions waiting for miscellaneous logs.
Log Synchronization Wait	Time spent by the various sessions waiting to see whether log truncation frees log space.
Log Wait	Time spent waiting by the various processes waiting for a Log operation to complete.
Log Write Wait	Time spent by the various sessions waiting for outstanding I/Os to finish, or waiting for log flushes to complete.
Memory Wait	Time spent by the various sessions waiting for memory resources to be made available.
Network HTTP Wait	Time spent by the various sessions waiting for outstanding HTTP connections to complete and exit.
Network I/O Wait	Time spent by the various sessions waiting for network packets.
Network IPC Wait	Time spent by the various sessions waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.
Network Mirror Wait	Time spent by the various sessions waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.
Network Wait	Time spent by the various sessions waiting for network I/O operations to complete.
Network Wait Percent	Percentage of time spent by the various sessions waiting for network I/O operations to complete.
Non CLR Wait	Overall active time that does not include CLR wait time.
Non CLR Wait Rate	The average number of active sessions waiting for all wait events, excluding CLR wait, during the specified time range.
Non CPU Wait	Overall active time that does not include CPU wait time.
Non CPU Wait Rate	The average number of active sessions waiting for all wait events, excluding CPU wait, during the specified time range.
Non I/O Wait	Overall active time during the specified time range, excluding time spent on I/O-related wait events.
Non Latch Wait	Overall active time during the specified time range, excluding time spent on latch-related wait events.
Non Latch General Wait	Overall active time during the specified time range, excluding time spent on general latch-related wait events.
Non Lock Wait	Overall active time during the specified time range, excluding time spent on lock-related wait events.
Non Log Wait	Overall active time during the specified time range, excluding time spent on log-related wait events.
Non Memory Wait	Overall active time during the specified time range, excluding time spent on memory-related wait events.
Non Network Wait	Overall active time during the specified time range, excluding time spent on network-related wait events.

Display Name	Description
Non Other Wait	Overall active time during the specified time range, excluding time spent on general other wait events.
Non Remote Provider Wait	Overall active time during the specified time range, excluding general remote provider wait time.
OLEDB Provider Full Text Wait	Time spent by various processes waiting for the Microsoft Full Text Engine for SQL Server.
Other Miscellaneous Wait	Time spent by the various sessions waiting for miscellaneous database operations.
Other Wait	Miscellaneous instance waits consisting of infrequent or otherwise special purpose wait states that should, in most cases, be very close to 0.
Parallel Coordination Wait	Time spent by the various sessions waiting for parallel coordination tasks to complete. This is the time spent by the various processes coordinating parallel query threads and exchanging data.
Remote Provider Wait	The time spent by the various processes waiting for a remote OLEDB call to complete or DTS synchronization.
Service Broker Wait	Time spent by the various sessions waiting for Service Broker event handlers and endpoints.
Synchronous Task Wait	Time spent by the various sessions waiting for tasks started synchronously (most SQL Server processes are started asynchronously).
XTP Transaction Wait	Time spent waiting for in-memory transaction events.
XTP Wait	Time spent waiting for in-memory OLTP (XTP) related events.
XTP Miscellaneous Wait	Time spent waiting for in-memory miscellaneous events. Those events are not related to transactions, log or natively compiled stored procedures.
XTP Log write Wait	Time spent waiting for in-memory log and checkpoint related events.
XTP Procedure Wait	Time spent waiting for events related to natively compiled stored procedures.

Instance Wait Events

Purpose

This collection provides data for a list of SQL Server wait events, divided by wait events.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Wait Event Name	Internal wait event name.
Wait Time	Total wait time for this wait event type, in seconds.

Job Messages

This collection provides a list of the job messages that were displayed as a result of running a specific SQL Server job.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Message	Text, if any, of a SQL Server error.
Step Name	The name of the step.
Step ID	The ID of the step in the job.
SQL Message ID	ID of any SQL Server error message, which is returned if the job failed.
SQL Message Severity	The severity of any SQL Server error.
Run Status	Status of the job execution. The run status can have one of the following values: • Failed • Success • Retry • Canceled • In progress
Message Time	The date and time when the job or step started execution. For an In Progress history, this is the date and time when the history was written.
Run Duration	The time, in seconds, which has elapsed since the beginning of the execution of the job or step format.

Latches and Locks

Purpose

This collection provides statistic details about latches and locks, such as wait time, requests and occurrences.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Average Latch Hit Wait Time	The wait time of latch requests that were satisfied.
Average Lock Duration	Average duration of blocked lock requests (100 * Lock Wait) /Blocked Lock Requests.
Average Lock Requests Wait Time	
Average Lock Wait Time	The average wait time, in milliseconds, of lock requests that were satisfied.
Blocked Lock Requests	Number of lock requests that could not be satisfied immediately, causing the caller to block and wait before acquiring the lock.
Blocked Lock Requests Ratio	The percentage of lock requests that required the caller to wait before acquiring the lock. A high percentage (over 20 percent), along with high Lock Wait, indicate that the system is not handling locks well and concurrency mechanisms need to be tuned. Excessive blocking can be a major cause of poor application performance, as the user of an application often does not realize that they are waiting on a lock held by another user. From their point of view, it often seems like their application has stopped responding. (100 * Blocked Lock Requests) / Lock Requests
Latch Wait Count	Number of latch requests that could not be satisfied immediately, causing the caller to block and wait before acquiring the latch.
Lock Requests	Number of new locks and lock conversions requested from the lock manager.
Lock Timeouts	The number of lock timeouts. By default, SQL Server never times out locks. However, many applications issue a SET LOCK_TIMEOUT statement to cause SQL Server to time out their locks after the specified interval. This metric shows how often these timeouts are being exceeded.
Lock Timeouts Bigger Than Zero	The number of lock timeouts. By default, SQL Server never times out locks (they will wait forever). However, many applications issue a SET LOCK_TIMEOUT statement to cause SQL Server to timeout their locks after the specified interval. This metric shows how often these timeouts are being exceeded.
Lock Wait Time	Total wait time, in milliseconds, for all lock requests that resulted in a wait.
Number Of Deadlocks	The number of lock requests that resulted in a deadlock. A Deadlock occurs when multiple SQL Server sessions request conflicting locks in such a way that two locks are blocked by one another.

Display Name	Description
Number Of Super Latches	Number of latches that are currently SuperLatches. Super Latches (or Sub-Latches) are latches that are generally used when data is accessed frequently (latches that are in shared lock mode (SH)). A regular latch is often “Promoted” to a Super/Sub latch in order to reduce the latch acquired on each lookup. Thus, when data is accessed many times for carrying out a read-only operation, the “newly promoted” Super/Sub latch will handle all requests for this data. An example to a Super/Sub latch would be an index root page accessed frequently.
Super Latch Demotions	Number of SuperLatches that have been demoted to regular latches. A latch is demoted when SQL Server determines that no thread requests a data help by Super/Sub latch, or when SQL Server needs to write to that data resource and need to acquire exclusive access to that source.
Super Latch Promotions	Number of latches that have been promoted to SuperLatches. A regular latch is often “Promoted” to a Super/Sub latch in order to reduce the latch acquired on each lookup. Thus, when data is accessed many times for carrying out a read-only operation, the “newly promoted” Super/Sub latch will handle all requests for this data. An example to a Super/Sub latch would be an index root page accessed frequently.
Total Latch Wait Time	The total amount of time, in milliseconds, that latch requests spent waiting during the last second.

Lock Statistics

Purpose

This collection provides detailed statistics for each lock.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Deadlocks	The total number of deadlocks for the lock type. A Deadlock occurs when multiple SQL Server sessions request conflicting locks in such a way that two locks are blocked by one another.
Lock Requests	The summary of all lock requests, both for new locks and lock conversions.

Display Name	Description
Lock Timeouts	The number of lock requests per second that timed out, including requests for NOWAIT locks.
Lock Type	The name of the lock type.
Total Wait Time (seconds)	The total lock wait time for the lock type.
Total Waits	The total number of wait events for the selected lock type.

Locks List

This collection provides a list of locks used within a SQL Server instance, grouped by the lock type, mode, and status.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60

Collection Metrics

Display Name	Description
SPID	The server process ID of the current user process.
Login Time	The time the session was created.
Login Name	SQL Server Login name.
Database	The database that is being used by the current process.
Index Name	The index that is being used by the current process (if any).
Object Name	The name of the currently locked object.
Count	The number of locks that share the same object and type.
Lock Type	The type of the resource that is currently being locked (Database, Table, Page, Row, Extent and so on).
Mode	The kind of lock that is being applied to the resource (Shared, Exclusive, Update, IntentShared, IntenExclusive and so on).
Status	The status of the lock can be one of the following: • WAIT = wait for another resource • CNVT = converted to another resource • GRANT = the lock has granted

Log Shipping

This collection specifies the Log shipping tree and the primary and secondary servers' file location and operations.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20

Collection Metrics

Display Name	Description
SPID	The SQL process ID of the process that ran the query.
Rowset Key	The row identifier.
Owner	The current primary server.
Server Name	The name of the server (primary server for the parent node, and secondary server for all subsequent child nodes).
Database Name	Refers to either of the following databases: • For the parent node, this is the database that is backed up on the source server. • For each child node(s), this is the destination database on the secondary server.
Flag Error	An indicator of an error that occurred.
Threshold	The maximum allowed time (in minutes) before a Log Shipping alert occurs. This parameter can refer to either of the following values: • For each parent node, this is the maximum allowed time since the last transaction log backup was made on the source server. • For each child node, this is the maximum allowed time between the last backup on the source server and the last restore on the secondary server.
Threshold Alert	The alert displayed in the Error Log drilldown if the threshold is exceeded.
Threshold Alert Enabled	The alert displays only if it is enabled.
Last Change	This parameter can refer to either of the following values: • For the parent node, this is the last time that the database was backed up. • For the child node(s), this is the last time a file was copied or restored. The Activity Type field displays this information.
Last Change Type	The type of the activity that resulted in the log shipping's most recent change of state (Backup, Copy or Restore).
Last Backup Filename	The name of the file that was most recently backed up.
Last Backup Last Updated	The time at which the most recent backup operation was carried out.
Outage	During an outage, no alerts will occur when thresholds are exceeded (for both the parent and child nodes). If this field is blank, no outage will take place.

Display Name	Description
Backup In Sync	Indicates either of the following backup statuses: • The backup is in sync mode. • The backup is out of sync and needs to be synchronized.
Backup Delta	The last time a log transaction backup occurred in the primary server.
Last Copied Filename	The last backup file that was copied to the secondary server.
Last Copied Last Updated	The time when the backup file was most recently copied from the primary to the secondary server.
Last Loaded Filename	The name of the backup file that was most recently restored to the secondary server.
Last Loaded Last Updated	The time when the backup file was most recently restored to the secondary server.
Copy Delta	The date when the transaction log backup file was most recently copied to the secondary server.
Copy Enabled	Indicates whether copying is enabled.
Load Enabled	Indicates whether loading is enabled.
Load in Sync	The load synchronization can have either of the following statuses: • The load is in sync mode. • The load is out of sync and needs to be synchronized.
Load Delta	The date of the last restored transaction log load file to the secondary server.
Primary Server Time	The current timestamp at the primary server.
Secondary Server Time	The current timestamp at the secondary server.
Monitor Server Time	The current timestamp at the monitored server.

Log Shipping Error

The collection shows Log Shipping Errors.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	900

Collection Metrics

Display Name	Description
Session ID	The session ID for the backup/copy/restore/job.
Sequence Number	An incremental number indicating the correct order of information for errors that span multiple records.
Log Time	The date and time at which the record was created.
Server Name	The name of the server.
DB Name	The name of the database.
Agent Type	The type of log shipping job. 0 = Backup. 1 = Copy. 2 = Restore.
Agent Type Description	The description of the Agent Type.
Message	Message text.
Source	The source of the Error Message.
Status	If no Error for last hour - OK. If an error found on the last 15 minutes - ERROR. If there is an error between last hour and last 15 minutes nothing would be raised so the last error status should remain.

Logical Disks

Purpose

This collection displays the logical disk activity for each logical disk.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Disk Queue Length	Disk Queue length tracks the average number of I/O requests, from all disks in the system, which are queued and waiting for a disk during a sampled interval. This figure may include I/O activity generated by processes other than SQL Server. Values that exceed the threshold set in this metric indicate a bottleneck.
Disk Queue Length Threshold	A flag indicating a threshold for the number of I/O requests that were outstanding on the busiest disk in the system.
Disk Percentage Used	The percentage of used space on the specified partition.
Disk Partition	The logical disks to which the Disk Queue Length value refers.
Partition Free Pct	The percentage of free space on the specified partition.
Partition Free Space	The amount of free space (in megabytes) on the specified partition.
Partition Total Space	The total space (in megabytes) on a particular partition.

Long Running Session

Purpose

The Long Running Session collection collects long-running query that runs more than specific time.

Collection Type

Oracle

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	120
Online	120
Offline	120

Metric Descriptions

Metric	Description
Session ID	Session identifier
Serial Number	Session serial number
SQL ID	SQL ID
Container ID	Container Name
Elapsed Time	Elapsed time in seconds since the session has become active.
Oracle User Name	Oracle user name
OS User	Operating system client user name
Program	Operating system program name
Machine	Operating system machine name

Metric	Description
Module	Name of the currently executing module
CPU Time	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends.
SQL Text	SQL Text

Memory Manager

Purpose

This collection monitors how SQL Server stores and manages its memory.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
% Total Server Memory	The total percentage of dynamic memory that the server is currently consuming. This value is calculated as follows: 100 * (Total Server Memory / Physical RAM)
Cache Memory	Total amount of dynamic memory (in megabytes) that the server is using for the dynamic SQL cache.
Connections Memory	Memory (in megabytes) allocated to keeping track of each connection's attributes.
Locks Memory	Memory (in megabytes) allocated to keeping track of locks.
Max SQL Server Memory	The maximum amount of dynamic memory that the instance is allowed to consume.
Memory Growth Pressure	Relative amount of additional memory (presented as a percentage) that SQL Server is willing to consume. Values higher than 20 percent might indicate that the system is low on physical memory. The memory growth pressure value is calculated as follows: 100 – ((100 * Total Server Memory) Target Server Memory)
Optimizer Memory	Work area for the Optimizer in megabytes.
Sort Hash Memory	Memory in megabytes used for the Sort, Hash, and Index operations.

Display Name	Description
Target Server Memory	The maximum memory allocation size in megabytes SQL Server can reach at a current time (this value may change during SQL Server's work). Values significantly higher than Total Server Memory might indicate that the system is low on physical memory.
Total Server Memory	The total amount of dynamic memory (in megabytes) that the server is currently consuming.

Mirroring

This collection retrieves general data, such as role, state and status, about each database involved in the mirroring process.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	900

Collection Metrics

Metric Display Name	Description
Connection Timeout	Defines the number of seconds that the principal database can be in Disconnected state, before the mirroring session will actually run a failover scenario to the mirroring database.
Database Name	The name of the database.
End Log LSN	The actual LSN (Log Sequence Number) of the mirroring session, This value should be equal to the Failover LSN value.
Failover LSN	The LSN (Log Sequence Number) value that the mirroring database should have in order to enable a smooth failover.
Mirror	The name of the instance whose role in the mirroring process is Mirror.
Mirroring Role	The role the database takes in the mirroring process; either principal or mirror.
Mirroring Status	Indicates the severity determined based on the database state: Normal, Warning or Critical.
Partner Name	The server network address defined for the Partner.
Principal	The name of the instance whose role in the mirroring process is Principal.
Redo Queue	The redo queue size. This size can be either lett Unlimited or defined in MB.
Replication LSN	The LSN (Log Sequence Number) used by the replication session (If such a session exists).
Role Sequence	A cumulative number that indicates the number of role switches that took place between the two partners involved in the mirroring process.

Metric Display Name	Description
Safety Level	The safety level at which the mirroring session is configured to work (high availability /high performance).
Witness Name	The server network address defined for the Witness.
Witness State	The state of the Witness instance (valid only when the High Safety level is configured).

Mirroring Performance Counters

This collection retrieves data regarding performance and data flow-related issues between the databases involved in the mirroring operation.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	900

Collection Metrics

Display Name	Description
Commit Acknowledgment Delay	Indicates a delay in waiting for acknowledgement of unterminated transaction commit. This metric is specific to the principal database, and holds values only when a Full safety level is configured.
Database Name	The name of the database.
Log Remaining For Undo	Indicates the size of transaction logs in the mirror database that remain for recovering the mirror database to its operative state. This metric holds data only after a failover has occurred.
Log Cache Redone	The amount of transaction log in MB that is being read from redo cache rather than from the transaction log. Constantly low values of this metric indicate that the transaction log is arriving faster than it can be read by the redo task. This metric is specific to the mirror database.
Log Harden Wait Time	The amount of time spent waiting for the log to be written to the mirroring database. High values of this metric can indicate that the disk of the mirroring database is loaded. This value is specific to the mirror database.
Log Received	Indicates the rate in MB of log received from the principal database. This metric is specific to the mirror database.
Log Rolled Forward	The size in MB of log that was rolled forward on the mirror database. This value is specific to the mirror

Display Name	Description
Log Scanned For Undo	Indicates the size of transaction logs in the mirror database that were scanned for Undo. This metric holds data only after a failover has occurred, as the log scanning is carried out in order to restore the mirror database to its operative state.
Log Send Flow Buffer Wait	The amount of time the mirroring session had to wait to use the mirroring flow control buffer. This value is specific to the principal database.
Log Sent	Indicates the rate in MB of log sent from the principal to the mirror database. This metric is specific to the principal database.
Log Sent From Cache	Indicates the amount of log, in MB, sent from the principal database cache rather than straight from the transaction log. As sending from the cache is much faster and more efficient, this metric's values should be as high as possible. Constantly low values indicate that the transaction log generation rate is faster than the rate at which it is sent to the mirror database. This value is specific to the principal database.
Mirroring Roundtrip (Latency)	Indicates the latency of the mirroring session.
Occupied Cache Percent	The percentage of the occupied cache within the selected database.
Roll Forward Queue	Total size in MB that remains to roll forward to the mirroring database. This value is specific to the mirror database.
Send Queue	Total size in MB of data waiting to be sent to the mirroring database. This value is specific to the principal database.
Write Commit	This metric, which holds values only when a Full safety level is configured, indicates the number of transactions in the principal database that had to wait for a write commit in the mirror database's transaction log. Low values of this metric indicate a bottleneck.

Missing Indexes

Collecting information about the indexes to be missing from each database.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	60
Offline	3600

Collection Metrics

Display Name	Description
Database ID	The database ID where the missing index is located.
Database Name	The database name.
Schema Name	The schema name of the missing index.
Object ID	The object ID where the missing index is located.
Object Name	The object name the resource is part of.
Object Type	Type of object.
Is Ms Shipped	Object is created by an internal SQL Server component.
Create Date Utc	Date the object was created.
Modify Date Utc	Date the object was last modified by using an ALTER statement. If the object is a table or a view, modify_date also changes when a clustered index on the table or view is created or altered.
Index Group Handle	Identifies a missing index group.
Index Handle	Identifies a missing index that belongs to the group specified by index_group_handle.
Group Handle	Identifies a group of missing indexes. This identifier is unique across the server. The other columns provide information about all queries for which the index in the group is considered missing. An index group contains only one index.
Unique Compiles	Number of compilations and recompilations that would benefit from this missing index group. Compilations and recompilations of many different queries can contribute to this column value.
User Seeks	Number of seeks caused by user queries that the recommended index in the group could have been used for.
User Scans	Number of scans caused by user queries that the recommended index in the group could have been used for.
Last User Seek Utc	Date and time of last seek caused by user queries that the recommended index in the group could have been used for.
Last User Scan Utc	Date and time of last scan caused by user queries that the recommended index in the group could have been used for.
Average Total User Cost	Average cost of the user queries that could be reduced by the index in the group.
Average User Impact	Average percentage benefit that user queries could experience if this missing index group was implemented. The value means that the query cost would on average drop by this percentage if this missing index group was implemented.
System Seeks	Number of seeks caused by system queries, such as auto stats queries, that the recommended index in the group could have been used for.
System Scans	Number of scans caused by system queries that the recommended index in the group could have been used for.
Last System Seek Utc	Date and time of last system seek caused by system queries that the recommended index in the group could have been used for.
Last System Scan Utc	Date and time of last system scan caused by system queries that the recommended index in the group could have been used for.
Average Total System Cost	Average cost of the system queries that could be reduced by the index in the group.
Average System Impact	Average percentage benefit that system queries could experience if this missing index group was implemented. The value means that the query cost would on average drop by this percentage if this missing index group was implemented.

Display Name	Description
Equality Columns	Comma-separated list of columns that contribute to equality predicates of the form: table.column =constant_value
Inequality Columns	Comma-separated list of columns that contribute to inequality predicates, for example, predicates of the form: table.column > constant_value Any comparison operator other than "=" expresses inequality.
Included Columns	Comma-separated list of columns needed as covering columns for the query. For more information about covering or included columns, see Create Indexes with Included Columns. For memory-optimized indexes (both hash and memory-optimized nonclustered), ignore included_columns. All columns of the table are included in every memory-optimized index.

Plan Cache Distribution

Details the plan cache distribution, grouped by plan types and their use of related statistics.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Plan Cache Name	The types of objects that can appear in the Procedure Cache include: Trigger Plans: Query plans generated by creating a trigger. Replication Procedure Plans: Query plans of a replication system stored procedure. Procedure Plans: Query plans generated by creating a stored procedure. Prepared SQL Plans: Query plans corresponding to statements prepared via sp_prepare, sp_cursorprepare, or via auto-parameterized or user-parameterized queries. Misc. Normalized Trees: Normalized trees for views, rules, computed columns, and check constraints. Execution Contexts: Data structures that hold the data (such as parameter values) that is specific to the execution of user queries. Cursors: Extensions to result sets that provide the mechanism for working with individual rows, or a small block of rows, in a table. Ad Hoc SQL Plans: Query plans produced for an ad hoc query. Types of objects that can appear in the Procedure Cache for a SQL Server 2005 connection include: Bound Trees: Normalized trees for views, rules, computed columns, and check constraints. Extended Stored procedures: Catalog information for extended store procedures. Object plans: Query plans generated by creating a stored procedure, function, or trigger. SQL Plans: Query plans corresponding to statements prepared via sp_prepare, sp_cursorprepare, or via auto-parameterized or user-parameterized queries. Temporary Tables & Table Variables: Cache information related to temporary tables and table variables.
Cache Size	The amount of memory allocated to the procedure cache over time and breaks it down by object type.
Cache Hit Rate	The amount of memory allocated to the procedure cache over time and breaks it down by object type.
Cache Reads	Cache lookups.
Cache Hit Rate Percent	The percentage of compilations that are being saved by objects already in the procedure cache. The hit rate for this cache will vary widely depending on how your applications are written.
Object Count	The number of objects of each type that are currently in the procedure cache.
Use Count	The rate at which each type of procedure cache objects are being executed (used).

Plan Cache List

Purpose

This collection provides a summary of SQL Server batches and compilations.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60

Collection Metrics

Display Name	Description
Allocated Size	The amount of space in the procedure cache (in megabytes) that is allocated to this object.
Cache Object Type	The type of object in the cache. This metric can have one of the following values: • Compiled Plan • Executable Plan • Parse Tree • Extended Stored Procedure
Database	The database where the object resides.
From Cache	Percent of memory consumed by the cache object as part of the whole plan cache memory.
Object	The object name.
Object Type	An object can have one of the following types: • Stored procedure • Prepared statement • Ad hoc query1 • ReplProc • Trigger • View • Default • User table • System table • Check • Rule
Owner	The object owner.
Reference Count	The number of other cache objects that reference this cache object.
SQL Text	The name of the procedure, or the first 128 bytes of the batch submitted.
SQL Text Size	Object size in bytes.
Use Count	The number of times this cache object has been used since inception.
Used Date Format	The date format used by the connection that created the cache object.
Used Language	The language of the connection that created the cache object.

Replication Agents

Purpose

This collection provides detailed data regarding all replication agents of the selected distributing SQL Server instance (Distributor).

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	300

Collection Metrics

Display Name	Description
Agent Name	The name of the replication agent.
Average commands	Average number of commands per transaction delivered in the session.
Delivery Latency	Latency, in milliseconds, between the transaction entering the distribution database and being applied to the Subscriber.
Delivery Rate	Average number of commands delivered per second.
Duration	The elapsed time of the agent's last session.
Publication DB Name	The name of the publication database, whose data is replicated using the selected agent.
Publication Name	The name of the publication replicated using the selected agent.
Publishing Server	The name of the SQL Server instance that hosts the publication database, whose data is replicated using the selected agent.
Start Time	The time when the selected session started.
Status	The selected agent’s current status; for example, Started, Completed, Running, Idle, Retrying, and Error.
Subscribing Server	The name of the SQL Server instance that hosts the Subscription database to which data is replicated using the selected agent.
Subscription DB	The name of the Subscription database to which data is replicated using the selected agent.
Subscription Type	Type of subscription; for example, Push, Pull, and Anonymous.
Type	The type of the selected agent; for example, Distribution, Log Reader, Snapshot, Merge, Miscellaneous.

Replication Agent Session Actions

Purpose

This collection provides detailed data regarding the actions carried out by the selected session of the replication agent.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	300

Collection Metrics

Display Name	Description
Agent Name	The name of the replication agent.
Message	All informational messages and error messages that the selected agent has logged during the selected session.
Session Start Time	The time when the selected session started.
Time	The time at which the action described in the Message column was performed.

Replication Agent Session Merge Articles

Purpose

This collection provides a detailed list of all articles replicated by the Merge agent's selected session.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	300

Collection Metrics

Display Name	Description
Article Name	The name of a specific article (a table or other object) that is part of the publication.
Counter for Conflicts	The total number of conflicts detected when trying to replicate the selected article.
Counter for Deletes	The total number of Delete operations that were replicated.
Counter for Inserts	The total number of Insert operations that were replicated.
Counter for Retries	The total number of retrial attempts performed when trying to replicate this article.
Counter for Schema Changes	The total number of schema changes detected.

Display Name	Description
Counter for Updates	The total number of Update operations that were replicated.
Duration	The amount of time the article replication took.
Merge Agent Name	The name of the selected Merge agent.
Percent of Total	The currently completed percent of the article replication operation.
Session Start Time	The time when the selected Merge session started.
Status	The Status of the Merge agent. The possible values are: Started, Completed, Running, Idle, Retrying, and Error.

Replication Agent Sessions

Purpose

This collection provides a detailed list of the selected agent’s closed sessions.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	300

Collection Metrics

Display Name	Description
Actions	Number of actions (comments) written per session.
Agent Name	The name of the selected agent.
Commands	The aggregated number of commands that were processed during the selected session's run.
Duration	Amount of time the agent has run in this session. The time represents elapsed time if the agent is currently running and the total time of the session if the agent session has ended.
End Time	The time the selected non-active session ended.
Last Message	All informational and error messages logged by the agent during the selected session.
Session Start Time	The time the selected non-active session started.
Status	Status of the agent session; for example, Started, Completed, Running, Idle, Retrying, and Error.
Transactions	The aggregated number of transactions that were processed during the selected session's run.
Type	The replication agent type.

Replication Agent Sessions by Type

Purpose

This collection provides a detailed list of the selected agent’s active sessions by type.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Actions	Number of actions (comments) recorded per session.
Agent Name	The name of the selected agent.
Commands Processed	The aggregated number of commands that were processed during the selected session's run.
Duration	Amount of time the agent has run in this session. The time represents elapsed time if the agent is currently running and the total time of the session if the agent session has ended.
End Time	The time the selected active session ended.
Last Message	All informational and error messages logged by the agent during the selected session.
Start Time	The time the selected active session started.
Status	Status of the agent session; for example, Started, Completed, Running, Idle, Retrying, and Error.
Transactions Processed	The aggregated number of transactions that were processed during the selected session's run.
Type	The replication agent type.

Replication Available

Purpose

This collection indicates the status of the SQL Server replication service.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20

Frequency Mode	Collection Interval (In Seconds)
Online	60
Offline	300

Collection Metrics

Display Name	Description
Replication Available	Indicates whether replication is available.

Replication Publications

Purpose

This collection provides detailed data about the publications replicated by the selected distributing SQL Server instance (Distributor).

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	900

Collection Metrics

Display Name	Description
Due to Expire	Displays the time and date when the subscription expires if changes recorded in the distribution database are not yet synchronized with the subscribing server.
Errors	Number of errors encountered when synchronizing with the publication.
Last Synchronized	Retention period of the publication, in hours. Indicates the predefined distribution retention period. If by the end of this period there are changes in the distribution database that have not been delivered to the subscribing server and synchronized, the subscription expires and has to be re-initialized.
Log Reader Agent	Name of the Log Reader agent, if applicable.
Publication DB Name	The name of the database that contains the selected publication.
Publication Name	The name of the currently selected publication.
Publishing Server Name	The name of the SQL Server instance that hosts the selected publication database.
Queue Reader Agent	Name of the Queue Reader agent, if applicable.

Display Name	Description
Snapshot Agent	Name of the Snapshot Agent job for the publication.
Status	Highest (worst) severity status of the agents associated with the publication. For example, Started, Succeeded, In progress, Idle, Retrying, and Failed.
Subscriptions	Number of subscriptions for the selected publication.
Synchronizing	Number of subscriptions currently being synchronized with the publication.
Type	The selected publication's type; for example, Transaction, Snapshot, or Merge.
Warnings	Number of publication monitor threshold warnings generated by the selected publication.

Replication Subscriptions

Purpose

This collection provides detailed data of the subscriptions of the selected distributing SQL Server instance (Distributor).

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	900

Collection Metrics

Display Name	Description
Distribution / Merge Agent	Name of the Distribution or Merge Agent for the subscription.
Latency	Highest latency, in seconds, for data changes propagated by the Log Reader or Distribution Agents for a transactional publication.
Log Reader Agent	Name of the Log Reader agent, if applicable.
Publication DB	The name of the database, which contains the publication that is replicated to the selected subscription
Publication Name	The name of the publication replicated to the selected subscription.
Publication Type	The type of publication replicated to the selected subscription; for example, Snapshot, Transactional, and Merge.
Publishing Server Name	The name of the SQL Server instance that hosts the selected publication database, whose data is replicated to the selected subscription.
Snapshot Agent	Name of the Snapshot Agent job for the publication.
Status	Highest (worst) severity status of the agents associated with the replication. For example, Started, Succeeded, In progress, Idle, Retrying, and Failed.

Display Name	Description
Subscribing Server	The name of the SQL Server instance that hosts the database where the selected Subscription resides.
Subscription DB	The name of the SQL Server instance that hosts the database where the selected Subscription resides.
Subscription Type	The selected subscription's type; for example, Push, Pull, Anonymous.

Reporting Services

Collecting reporting services server general information.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	60
Offline	60

Collection Metrics

Display Name	Description
Reporting Services Service Database Name	The name of the metadata database.
Total Reports Failed	Count of executions that were failed.
Total Reports Success	Count of executions that were success.

Resource Pool

The collection shows the usage of each resource pool.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Pool Name	Resource Pool name.
Used Memory	The amount of memory used by the resource pool.
Max Memory	The maximum amount of memory the resource pool can have. If significantly higher than target, it might indicate memory pressure.
Target Memory	The target amount of memory the resource pool is trying to gain. In case the value is significantly lower than maximum, it might indicate memory pressure.
Used Percentage	Percentage of memory used out of the overall memory allocated for the resource pool.
Out of Memory Count	The number of failed memory allocations.
Min Memory Percent	Minimum memory percentage consumed by the resource pool.
Max Memory Percent	Maximum memory percentage allowed by the resource pool.
Is Enabled	0 - The resource governor is disabled; 1 - the resource governor is enabled.

Session Data

This collection provides detailed data of the selected SQL Server session.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Active	Indicates whether the process is active.
Blocked By	Which SPID (if any) holds locks that this session is waiting on (Blocked By).
CPU Usage	The Total CPU time that this session has used since connecting to the SQL Server.
Cache Hit Ratio	The percentage, for this session, of buffer cache hit ratio, that is: file read operations that were satisfied by the file system cache without requiring any physical I/O. The value of this metric should be as high as possible.
Context Info	Provides context information for the current session.
Current Wait Time	The amount of time this session has been waiting. This shows 0 if the session is not currently waiting.
DB User	The SQL Server login name for this session.
Database	The name of the database where the session is running.

Display Name	Description
Error Number	Indicates if the most recent statement completed by this session returned an error number; corresponds to the @@Error global variable for this session. Can have two states: • Zero value - no error was returned • Positive value - an error was returned; the corresponding error text will be shown.
Host Name	The name of the Client host that is running the session.
Last Batch Time	The time the last batch started execution (Last Batch Time).
Last Command	The current or previous command executed.
Lock Timeout	The amount of time that this session waits for lock requests to be satisfied. Corresponds to the @@Lock_Timeout global variable for this session.
Logical Reads	Indicates the number of logical reads per second currently being performed by this session. Logical reads are database page I/O requests that were satisfied from the Buffer Cache and therefore did not have to perform disk reads.
Login Time	The time the session was created.
Memory	Memory is the number of pages in the procedure cache that are currently allocated to this process. A negative number indicates that the process is freeing memory allocated by another process.
Most Recent SQL	Shows the batch of SQL statements last executed or currently executing by the selected session.
Physical Reads	Shows the number of physical reads per second currently being performed by this session.
Physical Writes	Shows the number of physical writes per second currently being performed by this session.
Program	The Program that the user is running in order to access SQL Server.
SPID	The SQL process number.
Status	The session Status (runnable, sleeping, blocked and so on).
System Process	A flag indicating the session type: • Y= a system session • N = a non-system session.
Total I/O	The total amount of I/O operations (including background I/O) performed by the session from the time it connected to the SQL Server.
Transaction Count	The number of open transactions; corresponds to the session's @@TRANCOUNT value.

Session Trace

This collection traces over events that occurred in the currently monitored SQL Server instance.

Type

System.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
SPID	The session unique identifier (Session Process ID). The unique number that SQL Server has assigned to identify the selected session.
Start Time	The time when the class event started.
End Time	The time when the class event ended.
Event Class	The type of class that was traced.
Duration	The duration, in milliseconds, of the event class.
Text Data	The SQL text that was captured as a result of the trace event.
Reads	The number of logical disk reads carried out by the server on behalf of the event.
Writes	The number of physical disk writes carried out by the server on behalf of the event.
CPU	The amount of CPU time (in milliseconds) used by the event.
Database ID	ID of the current database.
Object Name	System-assigned ID of the object.
Index ID	ID for the index on the object affected by the event.

SQL PI Instance Statistics

SQL PI Instance Statistics.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	60
Offline	60

Collection Metrics

Display Name	Description
CPU Usage	CPU consumption of all the requests running on the monitored instance. Value can be larger than clock time if there is more than one CPU running.
CPU Usage Rate	Rate of CPU consumption of all the requests running on the monitored instance.
Num Executions	The number of statements executed during the current interval.
Num Executions Rate	Rate of statements executed during the current interval.
Total Logical Reads	The total number of logical reads operations performed by all requests running on the instance.

Display Name	Description
Total Logical Reads Rate	The average number of logical reads operations per second performed by all requests running in the instance.
Total Writes	The total number of logical writes operations performed by all requests running on the instance.
Total Writes Rate	The average number of logical writes operations per second performed by all requests running in the instance.
Total Physical Reads	The total number of physical reads operations performed by all requests running in the instance.
Total Physical Rate	The average number of physical reads operations per second performed by all requests running in the instance.
Row Count	The number of rows that have been returned to the client.
Tempdb I/O	Amount of time spent waiting for IO against Tempdb.
Tempdb GAM Wait	Amount of time spent waiting for a Global Allocation Map (GAM) page of Tempdb. GAM page tracks which extents have been allocated. The high amount of wait for an allocation page may indicate that the Tempdb is not configured according to best practice recommendations.
Tempdb SGAM Wait	Amount of time spent waiting for a Shared Global Allocation Map (SGAM) page of Tempdb. SGAM tracks which extents are being used as mixed (shared) extents. The high amount of wait for an allocation page may indicate that the Tempdb is not configured according to best practice recommendations.
Tempdb PFS Wait	Amount of time spent waiting for a Free Space (PFS) page of Tempdb. PFS tracks the allocation status of each page and approximately how much free space it has left. The high amount of wait for an allocation page may indicate that the Tempdb is not configured according to best practice recommendations.
Plan Recompilations	The number of times the plan of the statement was recompiled. For batch - this metric indicates the number of times at least one statement in the batch was recompiled.
Elapsed Time	Amount of time it took for the execution to end. This includes both active and idle time.
Degree of Parallelism	The maximum number of SQL Server threads assigned to the activity.
Granted Query Memory	The maximum amount of memory (KB) allocated to the execution of an activity.

SQL Server Connections Summary

This collection provides general information about all of the connections in SQL Server.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Active Connections Percent	The percentage of active connections as part of the user sessions.
All Active Connections	The number of all SQL active connections.
Background Sessions	The total number of background sessions.
Blocked Connections	The total number of blocked sessions.
Blockers Connections	The total number of blocking sessions.
Connections Running DBCC	The total number of sessions running a DBCC command.
Foreground Sessions	The total number of foreground sessions.
Internal Connections	The number of internal connections of the SQL Server instance.
Machine Connections	The number of Client computers that currently have at least one SQL Server session.
Max Block Time	The maximum duration (in seconds) a current process was blocked.
Max Connections Allowed	The maximum number of user connections that SQL server allows.
Max Threads Allowed	The maximum number of threads that SQL server is configured to handle.
SQL Active Threads	The total number of the instance's currently active threads.
SQL Executions	The total number of time that the selected statement executed during the specified time range. **NOTE:** This data is available only if StealthCollect is installed and configured
System Connections	The number of SQL Server system sessions.
Total Connections	The total number of SQL Server sessions, including user and system sessions.
User Active Connections	The number of non-system sessions that are actively processing in SQL Server or that are waiting on locks.
User Connections	The number of SQL Server user (non-system) sessions, excluding SQL Server Agent sessions.
User Inactive Connections	The number of non-system sessions that are not actively processing in SQL Server.
Worker Threads Used Percent	The percentage of active threads out of the maximum number of threads that SQL server is configured to handle.

SQL Server Global Variables

This collection provides general SQL Server parameters, such as I/O, networking, and database operational time.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
CPU Busy	The total time the SQL Server instance has spent working since it was last started. The result is displayed in CPU time increments, or “ticks”, and is aggregated for all CPU units, so it may exceed the actual elapsed time. **NOTE:** To convert the value to microseconds, multiply it by @@TIMETICKS.
CPU Idle	The total time the SQL Server instance has been idle since it was last started. The result is displayed in CPU time increments, or “ticks”, and is aggregated for all CPU units, so it may exceed the actual elapsed time. **NOTE:** To convert the value to microseconds, multiply it by @@TIMETICKS.
I/O Errors	The number of I/O errors encountered by SQL Server during the specified time range.
Instance Packet Errors	The number of packet errors encountered by SQL Server during the specified time range.
Instance Packets In	The total number of network packets that were being sent to SQL Server from client applications during the specified time range.
Instance Packets Out	The total number of network packets that were being sent from SQL Server to Client applications during the specified time range.
Instance Packets Total	The rate at which network packets are being sent and received from SQL Server to client applications.
Instance Packet Errors	The rate at which SQL Server is encountering network packet errors.
Total Read Rate	Total number of physical reads carried out by SQL Server since the instance was last started.
Total Write Rate	Total number of physical writes carried out by SQL Server since the instance was last started.

SQL Server Host

Purpose

This collection provides general details about the SQL Server host, such as the networking environment and the host's I/O and CPU consumption.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
CPU Name	The unique identifier of the CPU within all CPUs.
CPU Usage	Time spent by the various sessions that consume CPU cycles. This reading is taken directly from the operating system, rather than from SQL Server wait states.
Disk Utilization	The percentage of time the busiest disk spent serving system-wide I/O requests. This metric serves as a measure for the system I/O load. High values may indicate device bottleneck.
Free Virtual Memory	The current amount of virtual memory available in percent.
Individual CPU Usage	CPU utilization of a single processor.
Memory Max Percent	The maximum memory, in percent, which SQL Server can obtain from the host physical memory.
Network Bandwidth	The total network bandwidth used capacity of the specified network card in percent.
Network Interface	The name of the network interface.
Network Total Bandwidth	The total network bandwidth capacity of the specified network card.
Network Used Bandwidth	The total network bandwidth used capacity of the specified network card.
Physical Host	Physical name of the SQL Server instance host machine.
Physical Memory	The amount (in megabytes) of host physical memory.
Physical Memory Available	The amount (in megabytes) of physical memory available to applications.
Physical Memory Used	The amount (in megabytes) of physical memory used by applications.
Physical Memory Used Percent	The amount (by percentage) of physical memory used by applications.
Process CPU Utilization	Indicates the CPU use by a single process.

Display Name	Description
Run Queue Length	System average run queue. The CPU run queue is a holding area for threads and processes that require the CPU when the CPU is busy serving other processes. The run queue length is an indicator of whether the system has sufficient CPU resources for all the processes it executes. High values, along with high CPU utilization, indicate that the system requires faster or additional CPU units to handle the given load.
Total Virtual Memory	The current amount of total virtual memory.
Used Virtual Memory	The current amount (in megabytes) of used virtual memory.

SQL Server Load

Purpose

This collection displays general data about SQL operations on a SQL Server instance.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Attempted Connections	Number of attempted connections, either successful or unsuccessful since SQL Server was last started.
Session Logons	Number of sessions whose activity started during the current interval.
Session Logons Delta Rate Calculation	The rate per second of sessions whose activity started during the current interval.

SQL Server Services

This collection indicates the status of SQL Server-related services, mail provides, log shipping and clustering.

Type

SQL Server Services.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	3600

Collection Metrics

Display Name	Description
Analysis Services	The status of Analysis Services service.
Cluster Servers Down	Number of cluster nodes that are down or unreachable.
Distributed Transaction Coordinator	The status of Distributor Transaction Coordinator service.
Integration Services	The status of SQL Server Integration Services service.
Last Alert	Number of alerts raised by SQL Server since the specified time.
Last Job Failed	Number of failed jobs since the specified time.
Log Shipping	The status of Log Shipping.
MS Full Text Search	The status of MS Full Text Search service.
Non Preferred Node	Number of cluster nodes that are declared as preferred servers.
Reporting Services	The status of SQL Server Reporting Services service.
SQL Agent	The status of SQL Agent service.
SQL Server Active Directory Helper	The status of SQL Server Active Directory Helper (ADH) service.
SQL Server Browser	The status of SQL Server Browser Service.
SQL Server Mail	The status of SQL Server Mail service.
SQL Server VSS Writer	The status of SQL Server VSS writer Service.
State Changed	The time the service state was last changed.

SQL Server Throughput

This collection provides SQL Server throughput metrics of the modification made, in both size (MB) and number of rows, as a result of carrying out the following queries: select insert, update, and delete.

IMPORTANT:

IMPORTANT: Small tables (< 50 rows) are not included in any of this collection's metrics.

NOTE:

NOTE: While the Delete (MB), Insert (MB), Select (MB), and Update (MB) metrics handle the size in megabytes, the average Select rate, displayed in the chart, can appear in Kb/Sec when the rate values are extremely low.

Type

SQL Server Services.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	900
Offline	3600

Collection Metrics

Display Name	Description
Delete (MB)	The cumulative size (in megabytes) of I/O Delete operations performed on indexed and heaped tables in the selected SQL Server instance.
Insert (MB)	The cumulative size (in megabytes) of I/O Insert operations performed on indexed and heaped tables in the selected SQL Server instance.
Select (MB)	The cumulative size (in megabytes) of I/O Select operations performed on indexed and heaped tables in the selected SQL Server instance.
Update (MB)	The cumulative size (in megabytes) of I/O Update operations performed on indexed and heaped tables in the selected SQL Server instance.
Delete (Rows)	The cumulative size, in rows, of I/O Delete operations performed on indexed and heaped tables in the selected SQL Server instance.
Insert (Rows)	The cumulative size, in rows, of I/O Insert operations performed on indexed and heaped tables in the selected SQL Server instance.
Instance	The SQL Server instance name.
Select (Rows)	The cumulative size, in rows, of I/O Select operations performed on indexed and heaped tables in the selected SQL Server instance.
Update (Rows)	The cumulative size, in rows, of I/O Update operations performed on indexed and heaped tables in the selected SQL Server instance.

SQL Server Version Info

This collection provides general data about the SQL Server instance properties, such as identification of the database version number, along with the most recently installed service packs

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	900
Online	900
Offline	86400

Collection Metrics

Display Name	Description
Cluster	Indicates if the instance is taking part in a cluster environment.
Collation	The SQL Server instance collation.
Log Shipping	Indicates if Log Shipping is configured.
Mirroring	Indicates if mirroring is configured.
OS Version	The NT version of the computer that runs the SQL Server instance.
Physical Memory	The amount of physical memory on the SQL Server instance host machine.
Process ID	The process ID of the SQL Server instance host machine.
Processor Count	The number of processors on the SQL Server instance's host machine.
Processor Type	The processor type on the SQL Server instance's host machine.
Product Version	SQL Server instance product version.
Server Name	The server name of the SQL Server instance.
SP Version	The service pack version of the SQL Server instance.
SQL Edition	The SQL Server instance edition.
SQL Version	SQL Server instance version.
WindowsVersion	The Windows version of the computer that runs the SQL Server instance.

SSIS OS Statistics

It collects SSIS OS statistics metrics.

Type

WMI.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	30
Online	30
Offline	30

Collection Metrics

Display Name	Description
Working Set	Working Set is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed they will then be soft-faulted back into the Working Set before leaving main memory.
% Processor	% CPU of SSIS Out of the entire CPU usage.
Memory Utilization	The percentage of memory being used by the host of SSIS.

SSIS Summary

Gather availability information about the service.

Type

Mixed WMI +SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	60
Offline	60

Collection Metrics

Display Name	Description
Available	0. Service is unavailable. 1. Service is available.
Up Since	Used to find how long the instance is up and running virtualization.
Working Set	Working Set is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they will then be soft-faulted back into the Working Set before leaving main memory.
IOPs	Number of IO operations per minute.
% Processor	% CPU of SSIS Out of the entire CPU usage.

Display Name	Description
Bytes Read	The number of bytes read from all data sources in total.
Bytes Written	The number of bytes written to all data destinations in total.
Rows Read	The number of rows that a source produces in total. The number does not include rows read from reference tables by the Lookup transformation.
Rows Written	The number of rows offered to a destination in total. The number does not reflect rows written to the destination datastore.
Buffers Memory	The average amount of physical and virtual memory used by the data flow task.
Buffers Spooled	The total number of buffers written to disk during data flow tasks in the selected timeframe. Buffer is written to disk when the data flow engine runs low on physical memory.
Buffers In Use	The average number of buffer objects used by all data flow components and data flow engine during the selected timeframe.
Total Executions Count	The total number of packages execution in the examined era.
Failed Executions Count	The total number of failed packages.
Currently Running	The number of currently running packages.
Succeeded Execution Count	The number of successes packages run.
Others	Other packages running results.
Distinct Executed Packages	The number of distinct packages runs.
Current Number of Packages	The current number of packages.
Days Since Last Backup	Days since last backup.
Encryption Algorithm	The type of encryption algorithm that is used to encrypt sensitive data. The supported values include: DES, TRIPLE_DES, TRIPLE_DES_3KEY, DESX, AES_128, AES_192, and AES_256. **NOTE:** The catalog database must be in single-user mode in order to change this property.
Max Project Versions	The number of new project versions that are retained for a single project. When version cleanup is enabled, older versions beyond this count are deleted.
Operation Cleanup Enabled	When the value is *TRUE*, operation details and operation messages older than RETENTION_WINDOW (days) are deleted from the catalog. When the value is *FALSE,* all operation details and operation messages are stored in the catalog. **NOTE:** A SQL Server job performs the operation cleanup.
Retention Window	The number of days that operation details and operation messages are stored in the catalog. When the value is *-1*, the retention window is infinite. **NOTE:** If no cleanup is desired, set OPERATION_CLEANUP_ENABLED to FALSE.
Schema Build	The build number of the SSISDB catalog database schema. This number changes whenever the SSISDB catalog is created or upgraded.
Schema Version	The major version number of the SSISDB catalog database schema. This number changes whenever the SSISDB catalog is created or the major version is upgraded.
Server Logging Level	The default logging level for the Integration Services server.
Version Cleanup Enabled	When the value is *TRUE*, only the MAX_PROJECT_VERSIONS number of project versions are stored in the catalog and all other project versions are deleted. When the value is *FALSE*, all project versions are stored in the catalog. **NOTE:** A SQL Server job performs the operation cleanup.

Display Name	Description
Operation Log Number of Records	The number of total operations in the Integration Services catalog.
Operation Log Size	Total operation log size.
Version Log Number of Records	Total number of records in the log versions of objects in the Integration Services catalog.
Version Log Size	Total size of the objects versions log in the Integration Services catalog.

Statistics

Purpose

This collection provides a summary of SQL Server batches and compilations.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
Batch Executions	SQL Batches being submitted to SQL Server for execution.
Batch Executions Rate	The rate per second at which SQL batches are being submitted to SQL Server for execution.
Compiles	Compiling is the process by which SQL Server converts the SQL statement into an executable query plan.
Compiles Rate	The rate per second at which SQL Statements are compiled, that is: converted into an executable query plan.
ReCompiles	A recompile occurs when SQL Server determines that the execution plan for a stored procedure that is currently executing may no longer be the best possible plan.
Redundant Compilations Ratio	Percentage of SQL recompilations that take place. A high percentage of SQL recompilations may indicate the need for additional physical memory. This ratio is calculated as follows: 100 * (Compiles / Compiles + Recompiles)

Top SQLs

Purpose

This collection collects data of SQL Statements that consumed the highest amount of resources during the specified time range.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	900

Collection Metrics

Display Name	Description
Creation Date	Indicates the date the SQL statement was first run, thereby allowing the creation of SQL statement statistics
Diff. From Executions	Indicates the incremental difference in the number of SQL executions since the last sample.
Diff. From Total CLR Time	Indicates the incremental difference, since the last sample, in the total CLR time spent for executing the SQL statements.
Diff. From Total CPU	Indicates the incremental difference, since the last sample, in the total CPU time consumed by the SQL executions.
Diff. From Total Elapsed Time	Indicates the incremental difference, since the last sample, in the total time consumed for carrying out the SQL statement.
Diff. From Total I/O	The incremental difference, since the last sample, in the total number of I/O operations that were carried out for executing the SQL statements.
Diff. From Total Logical Reads	The incremental difference, since the last sample, in the total number of logical reads carried out for executing the SQL statements.
Diff. From Total Physical Reads	The incremental difference, since the last sample, in the total number of physical reads carried out upon executing SQL statements.
Diff. From Total Wait Time	The incremental difference, since the last sample, in the total wait time consumed upon executing SQL statements.
Executions	Indicates the number of times the SQL script executed for a unique SQL Handle.
Last Executed	Indicates the last time the SQL statement was executed (last run).
Plan Handle	Provides a unique identification of a SQL statement execution plan.
Query Hash	Query hash value indicates the logic behind a SQL script. Similar SQL queries (that is, SQL queries that differ by literal rather than logical values) have the same query hash value. Therefore, if the execution of such queries results in different query plan hash values, the reason for the difference is most probably the data against which the queries were executed.

Display Name	Description
SQL Handle	Provides a unique identification of a SQL batch script.
SQL Hit Rate	The percentage of SQL Statements that found the required plan already in the cache. If a matching plan is found, SQL Server does not need to compile the query or the stored procedure. This can save a significant amount of CPU resources, and speed up SQL Server queries.
Statement End Offset	A SQL batch may be combined of multiple SQL statements. While the SQL handle identifies the batch as a whole, this metric provides a unique identification of a specific SQL statement, by indicating the end position of the statement within the SQL batch.
Statement Start Offset	A SQL batch may be combined of multiple SQL statements. While the SQL handle identifies the batch as a whole, this metric provides a unique identification of a specific SQL statement, by indicating the start position of the statement within the SQL batch.
Total CLR Time	The total CLR time spent for executing the SQL statements.
Total CLR Time Per Exec	The average CLR time spent for executing a single CLR execution.
Total CLR Time Rate Top SQL	The total CLR time per second spent for executing the CLR execution.
Total CPU	The total CPU time consumed by the SQL executions.
Total CPU Per Exec	The average CPU time spent for executing a single SQL statement.
Total Elapsed Time	The total time consumed for carrying out the SQL statement executions.
Total Elapsed Time Per Exec	The average time spent for executing a single SQL statement.
Total I/O	The total number of I/O operations that were carried out for executing the SQL statements.
Total I/O per Exec	The average number of I/O operations carried out for executing a single SQL statement.
Total I/O Rate Top SQL	The total number of I/O operations per second that were carried out for executing the SQL statements.
Total Logical Reads	The total number of logical reads carried out for executing the SQL statements.
Total Logical Reads Per Exec	The average number of logical reads carried out for executing a single SQL statements.
Total Logical Reads Rate Top SQL	The total number of logical reads per second that were carried out for executing the SQL statements.
Total Physical Reads	The total number of physical reads carried out for executing the SQL statements.
Total Physical Reads Per Exec	The average number of physical reads carried out per execution of a single SQL statement.
Total Wait Time	The total wait time spent while executing the SQL statements.
Total Wait Time Per Exec	The average time spent on wait events while executing a single SQL statement.

Top SQL Batch Text

Purpose

This collection retrieves the full batch SQL text script of a given SQL handle.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
SQL Batch	The full text of the SQL execution batch.

Top SQL Long Text

Purpose

This collection retrieves the full SQL text script of a given SQL handle.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Query Text	The full text of the SQL statement.

Top SQL Plan

Purpose

This collection retrieves a query plan of a given SQL handle.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

IMPORTANT:

IMPORTANT: If the cache was flushed after the SQL statement was executed, the query plan cannot be retrieved.

Collection Metrics

Display Name	Description
Query Plan	The full query plan of the SQL statement.

Top SQL Short Text

Purpose

This collection retrieves part of a SQL text script of a given list of SQL handles.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Object Name	The name of the object owning the SQL; for example, view, procedure or function. This value can be Unavailable for SQL statements that are not object-owned, such as Ad-Hoc queries.
Query Text	The SQL execution text. The text's length is configurable, and its default value is 1000 characters.
Is Full Text	Indicates if all the retrieved text can be displayed in the short text field. This metric has a true value only for SQL statements whose text is shorter than the limit defined for short texts.

Top SQL Summary

Purpose

This collection Summarizes data from the Top SQL collection.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
CPU Usage	Time spent by the various sessions consuming CPU cycles. This reading is taken directly from the operating system, rather than SQL Server wait states.
CPU Usage Rate	The average time (seconds) spent by the various sessions consuming CPU cycles.
Executions	Indicates the number of times the SQL script per SQL Handle executed
Total CPU	The total CPU time consumed by the SQL executions
Total Physical Reads	The total number of physical reads carried out for executing the SQL statements
Total Logical Reads	The total number of logical reads carried out for executing the SQL statements
Total Elapsed Time	The total time consumed for carrying out the SQL statement executions
Total CLR Time	The total CLR time spent for executing the SQL statements
Total I/O	The total number of I/O operations that were carried out for executing the SQL statements
Total Wait Time	The total wait time spent while executing the SQL statements
SQL Hit Rate	The percentage of SQL Statements that found the required plan already in the cache. If a matching plan is found, SQL Server does not need to compile the query or the stored procedure. This can save a significant amount of CPU resources, and speed up SQL Server queries.

Traced SQL PA

Purpose

This collection retrieves information about the most recent statements executed by the selected session, including statements that have also already completed running. Each statement refers to a specific execution.

Type

SQL Server.

Collection Sampling Intervals

This collection is an on-demand collection, which has no pre-defined sampling intervals.

Collection Metrics

Display Name	Description
Active Time	Sum of all active wait events; equal to the session total activity within the specified time range.
CLR Wait	Time spent by the various sessions waiting for CLR code execution to complete.
CPU Usage	Time spent by the various sessions consuming CPU cycles. This reading is taken directly from the operating system, rather than SQL Server wait states.

Display Name	Description
CPU Wait	Time spent by the various sessions waiting in the systems run queue for CPU cycles. This reading is calculated from the operating system readings, rather than SQL Server wait states.
Database	The name of the database where the session is running.
Degree of Parallelism	Average number of SQL Server threads assigned to serve the current SQL statement.
End Time	The time when the selected statement's execution ended.
Event (list)	The list of wait events for the selected SQL statement.
Event Wait (list)	Time spent on the wait events for the selected SQL statement.
Instance CPU Usage	The amount of CPU time that the SQL Server instance consumed.
IO Wait	Time spent by the various processes waiting for the completion of physical I/O operations.
Latch Wait	Time spent waiting for internal locks (latches) to be released.
Lock Wait	Time spent by the various sessions waiting for a blocking lock (held by another session) to be released.
Log Wait	Time spent waiting by the various processes waiting for a Log operation to complete.
Memory Wait	Time spent by the various sessions waiting for memory resources to be made available.
Network Wait	Time spent by the various sessions waiting for network I/O operations to complete.
Other Wait	Miscellaneous instance waits consisting of infrequent or otherwise special purpose wait states that should, in most cases, be very close to 0.
Physical IO	The total amount of I/O operations carried out so far by the statement.
Remote Provider Wait	The time spent by the various processes waiting for a remote OLEDB call to complete or DTS synchronization.
SPID	The process ID of the blocking session.
SQL Text	The short text of the SQL statement.
Start Time	The time when the SQL statement started.

Usability

Purpose

This collection retrieves instance usability based on response time and instance availability.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60

Frequency Mode	Collection Interval (In Seconds)
Online	60
Offline	300

Collection Metrics

Display Name	Description
Availability	Indicates whether the SQL Server instance is available.
Instance Response Time	The full round-trip response time (in milliseconds) of a query representative of general workload (select 1, by default).
Instance Up Since	The SQL Server startup time.
Instance Up Time	The amount of time that the SQL Server instance was running during the current interval. If the instance is down, no data collection occurs. Therefore, if this metric value is less than the interval length, the value of this metric reflects only the active portion of the interval.
Instance Unavailable	The relative amount of time that the SQL Server instance was down during the current interval. If the instance is down, no data collection occurs.
Connection Time	The time when the connection was established.
OS Connect Availability	Indicates whether a connection is available.

User-defined Performance Counters

Purpose

This collection allows selecting which counters to monitor from the entire performance counters' list.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	300

User-defined Queries

Purpose

This collection allows creating SQL statements. Statements created using this collection should return only one value, the data type of which is numeric.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	300
Online	300
Offline	300

Virtualization

It collects Virtualization metrics from the ESX host.

Type

System.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	20
Online	60
Offline	300

Collection Metrics

Display Name	Description
CPU Ready Time	The amount of CPU time (in seconds), since the last measurement, that is unavailable to this virtual machine, because it is being utilized by either other virtual machines or the VMware itself.
CPU % Overhead	The percentage of CPU that is unavailable to this virtual machine because it is being utilized by either other virtual machines or VMware itself.
ESX CPU Usage	The amount of CPU actively used on the host, as a percentage of the host's total available CPU. Active CPU is approximately equal to the ratio between used and available CPU. This metric is calculated as follows: Available CPU = # of physical CPUs x clock rate.
ESX Machine Memory Size	The total amount of physical memory (in GB) on the ESX host machine.
ESX Memory Active	The amount of memory that is being actively used, as estimated by VMkernel based on recently accessed memory pages. This metric is calculated by summing all of the active metrics for all running (powered-on) virtual machines on the host, in addition to vSphere services.
ESX Memory Usage	The amount of memory used by the ESX host, as a percentage of the total amount of configured or available memory.

Display Name	Description
ESX Memory Balloon	The amount of memory allocated by the virtual machine memory control driver (vmmemctl), that is, the amount of guest physical memory currently reclaimed from the virtual machine through ballooning.
ESX Memory Shared	The total sum of guest memory that is shared between all powered-on virtual machines on a host. Shared memory, which refers to multiple references to a specific page held in memory, may be allocated to the virtual machine on top of the physical memory allocation (overcommitment). This metric's value reflects the efficiency of page sharing and overcommitment in utilizing machine memory.
ESX Memory Swap In	The amount of guest physical memory that was swapped-in to memory from disk via the Vmkernel, since the virtual machine was powered on. This metric is calculated as the sum of swapin values for all powered-on virtual machines on the host.
ESX Memory Swap Out	The amount of guest physical memory that was swapped out from memory to disk via the Vmkernel, since the virtual machine was powered on. This metric is calculated as the sum of swapout values for all powered-on virtual machines on the host.
ESX Memory Swapped	The total amount of guest physical memory that was swapped via the VMkernel, since the virtual machine was powered on, both to memory from disk (swapped in) and from memory to disk (swapped out). This figure, which refers to VMkernel swapping and not to guest OS swapping, is calculated as follows: esx_memory_swapin + esx_memory_swapout.
ESX Memory Swap Used	The total amount of swapped memory of all powered-on VMs and vSphere services on the ESX host.
VM Connect Availability	This is an indication of the virtualization collection connectivity. 0 - connection failed 100 - connection valid
VM Error Message	The error message indicating the reason for the collection failure to retrieve the data from the vSphere Services

XTP Session Transactions Statistics

Summary information from XTP Transaction statistics.

Type

SQL Server.

Collection Sampling Intervals

Frequency Mode	Collection Interval (In Seconds)
Real-time	60
Online	300
Offline	900

Collection Metrics

Display Name	Description
XTP Transactions	The total number of transactions that have run in the In-Memory OLTP database engine.
XTP Transactions Rate	The total number of transactions that have run in the In-Memory OLTP database engine per second.
XTP Read Only Transactions	The number of read-only transactions.
XTP Aborted Transactions	The total number of transactions that were aborted, either through user or system abort.
XTP Failed due to Validation	The number of times a transaction has aborted due to a validation failure.
XTP Failed due to Dependencies	The number of times a transaction aborts because of a transaction on which it was dependent aborts.
XTP Failed due to Unique Constraint	The total number of unique constraint violations.
XTP Failed due to Write Conflicts	The number of conflicts between transactions while writing a row which led to aborts.
XTP Writes (MB)	The total number of mega-bytes written to the In-Memory OLTP log records.
XTP Transactions on Durable Tables	The total number of transactions that require log I/O. Only considers transactions on durable tables.

6

Glossary

A

Access methods

A functional module within SQL Server that is responsible for managing the retrieval and modification of data and index pages.

Adhoc SQL Plans

Query plans produced from an ad hoc Transact-SQL query, including auto-parameterized queries. SQL Server stores such query plans in the plan cache for later reuse, if an identical Transact-SQL statement is later executed.

Alarm

The mechanism by which Foglight for SQL Server alerts to a condition that might be a problem in the SQL Server instance. While an alarm is active, the color of icons on the home page changes.

Alert

A SQL Server Agent object that performs an action when a condition is satisfied. Categorized as either Event Alerts or Performance Alerts. The action performed can be to send an e-mail, pager or Net Send message, or to run an SQL Agent job.

Allow updates

A SQL Server configuration parameter that determines if system tables can be modified using standard INSERT, UPDATE, DELETE statements. A value of 0 means updates are not allowed, and 1 indicates they are allowed.

Anonymous subscription

Anonymous subscription is a specific type of pull subscription, for which information regarding the subscription and the subscribing server (Subscriber) is not stored, for various reasons such as security (if the subscribing server accesses publications over the Internet) or cutting maintenance overhead involved in storing extra information at either the publication or distribution database. Anonymous subscriptions are initiated at the subscribing server, which is also responsible for keeping such subscriptions synchronized.

Authentication

The process of identifying and verifying a user who is attempting to establish a SQL Server session. Can be either Windows Authentication or SQL Server Authentication. With Windows Authentication (trusted logins), SQL Server uses Windows security mechanisms to determine who the users are and what they have access to. With SQL Server Authentication, users must have a SQL Server login and password, which are validated against a SQL

Server system table. SQL Server can be configured to allow only Windows Authentication, or both Windows and SQL Server Authentication (Mixed mode).

AutoClose

An option that can be set on SQL Server databases. This option causes SQL Server to periodically attempt to reorganize the data, thereby acquiring additional space. Continually opening and closing the SQL Server database significantly affects the database. Therefore, the AutoClose option is disabled by default, thereby keeping the database always open and ready, regardless of user activity.

AutoGrow

An option that can be set on SQL Server data and log files, to determine whether they can grow.

Automatic Discovery

Automatic Discovery is a cartridge that is installed as part of the Foglight for SQL Server installation package. This cartridge allows running the database discovery wizard, which provides a common entry point for adding and discovering all of the database instances within a user-specified range, and then configuring these instances to be monitored.

AutoShrink

An option that can be set on SQL Server databases, which causes SQL Server to periodically attempt to reorganize the data, thereby acquiring additional space or unused space from the file.

B

Batch

A set of Transact-SQL statements that are sent from a client program to SQL Server for execution. Even though batches can contain multiple statements, they often hold only a single statement.

BCP (Bulk Copy Program)

A stand-alone executable program that is extremely fast at copying data between SQL Server and text files. BCP can be used for loading or unloading data.

Blocking

The situation where a SQL Server session is waiting on a lock held by another session. The session waiting is said to be Blocked, and the one holding the lock is Blocking.

Books Online

SQL Server Books Online. The online help system that is provided with the SQL Server utilities.

Bound trees

Normalized trees for views, rules, computed columns, and check constraints.

Bound trees are used in SQL Server 2005 and later versions.

Buffer

A memory structure. Equates to a page that is currently held in memory (RAM).

Buffer cache

SQL Server's main memory cache. The buffer cache is an in-memory copy of recently used database pages, and contains database, free, and stolen pages.

The buffer cache's size is calculated by <free pages> * 8K / 1024.

Buffer pool

The buffer pool is the total buffer memory used by SQL Server.

When the SQL Server starts, it begins by computing the upper size of the buffer pool, which is the maximum size to which the Server allows the buffer pool to grow. The lower value between the maximal buffer pool size and the amount of physical RAM on the disk is the total amount of memory currently allocated to SQL Server.

If the dynamic memory management mechanism is properly tuned, SQL Server can automatically tune its memory allocations based on the load it is processing and the demands of other Windows processes on the Server.

Bulk copy

See BCP (Bulk Copy Program) on page 284.

Bulkinsert

A Transact-SQL statement designed to load data from a text file into a SQL Server table. This is the fastest way of getting data into SQL Server. The bulk insert effectively runs a BCP Load in the SQL Server address space.

Bulk load

See Bulk Insert.

C

Cache

An area of memory that holds a copy of data or objects. Because accessing information from RAM is faster than from disk, caching data in RAM improves performance by reducing disk I/O.

The SQL Server uses two types of cache storage: buffer cache, used for storing recently used database pages, and plan cache, which stores execution plans. For details, see glossary definitions of Buffer cache on page 285 and Procedure cache on page 301.

CAL

Client Access License. A license purchased from Microsoft, which allows a user to access SQL Server. For further details, see glossary definition of Licensing on page 295.

Calibration

The process by which Foglight for SQL Server determines the maximum and minimum values for every dataflow on the home page, by observing data moving through the database system. This information helps Foglight for SQL Server display the dataflows correctly.

Cardinality

A measure of the number of unique values within a column or index. The higher the cardinality of an index, the fewer the number of rows that are returned by an exact lookup of a key value (and hence the more useful the index).

Cartridge

Cartridges extend the functionality of Foglight and are installed on the Foglight Management Server. A cartridge contains one or more components, such as agents for deployment, communication capabilities, and modifications to the way that data is transformed or handled, as well as rules, reports, and views.

Chart

A graphical representation of a statistic over a period of time. One or more statistics may be shown on the same chart.

Checkpoint process

A SQL Server process that is dedicated to managing checkpoints. The Checkpoint process minimizes the amount of work SQL Server is required to do on restart, by periodically scanning the buffer cache for modified pages and writing all modified data out to disk.

Client network utility

An executable program installed with SQL Server. Assists in configuring which network libraries are used by client software that establishes a connection to SQL Server.

CLR

CLR (Common Language Runtime) is part of Microsoft's .NET Framework, which is integrated in SQL Server 2005 and later versions.

CLR provides the execution environment for all .NET Framework code, and allows using all of the available .Net languages, such as VB.NET or C#, for creating stored procedures, triggers, user-defined functions, user-defined types, and user-defined aggregates. Code that runs within the CLR, which is referred to as managed code, compiles to native code prior to execution, thereby helping to achieve significant performance increases in certain scenarios.

Compile

The process the SQL Server Query Optimizer uses for determining how a query or stored procedure is executed. During a compile, the Query Optimizer examines the query, including the tables, clause conditions, joins, sub queries, sort and grouping requirements used. The Optimizer then takes into account all of the existing indexes, locking methods and join algorithms that could help the query run faster. The result of this process is what the SQL Server Query Optimizer considers to be the fastest way of executing the query. This is known as the execution plan.

Connect

The process of establishing a link to SQL Server, including authorization checking. After a connection is established, a client program can access, modify, and store data in a SQL Server database.

Connection

See Session on page 304.

Connectivity software

A program used for establishing and maintaining a connection between a client program and SQL Server. The connectivity software usually used under SQL Server is Microsoft Data Access Components (see MDAC on page 297).

CPU Usage

When SQL statements and other types of calls are made to SQL Server, an amount of CPU time is necessary to process the call. Average calls require a small amount of CPU time. However, an SQL statement involving a large amount of data or a runaway query can potentially consume a large amount of CPU time, reducing CPU time available for other processing.

CPU utilization is the most important operating system statistic in the tuning process. Excessive CPU usage usually means that there is little idle CPU on the system. This could be caused by an inadequately-sized system, by untuned SQL statements, or by inefficient application programs.

Cursors

Extensions to result sets that provide the mechanism for working with individual rows, or a small block of rows, in a table.

Because a cursor points to a currently selected set of records, a cursor can be used by only one connection at a time. However, the compiled plan to which the cursor is linked can be used simultaneously by multiple connections.

D

Data access components

See MDAC on page 297.

Data file

A file that is one part of the “data” portion of a SQL Server database. Stores all data such as tables, rows, stored procedures, and indexes. While each database has one or more data files, each data file can belong to only one database.

Data storage engine

The major functional component of SQL Server, which parses and optimizes SQL requests, and processes results from the Storage Engine.

Database

A collection of tables and other objects. Each database contains a data portion and a log portion. The data portion contains all tables, rows, indexes and so on, and resides on at least one data file. The log portion tracks updates to the data, and aids recovery. Logs reside on one or more log files.

Database object

Anything that is stored in a database, such as Tables, Stored Procedures, Views, Triggers, User Defined Data Types, System Tables, and Constraints.

DataFlow

A Dataflow displays the current level of activity. As the rate of data transfer increases, so too does the speed of the flow. If the statistic represented by the flow moves to another threshold, the flow may change color. The combination of movement and color makes it easy to spot congested areas. A graph on top of the flow indicates how the load has varied over time.

DBCC

Database Console Command (sometimes called Database Consistency Checker). A SQL Server Transact-SQL statement that allows carrying out operations that are specific to SQL Server and are not covered by the ANSII SQL specification, such as checking the integrity of databases, retrieving database statistics and performance information, and viewing SQL Server internal information.

DBID

Database ID. A number that uniquely identifies a database within a SQL Server.

DBO

Database Owner. The user who owns the database.

Deadlock

A deadlock occurs when there is a cyclic dependency between two or more threads, or processes, for some set of resources within SQL Server, that is, each task has a lock on a resource which the other tasks are trying to lock.

When an instance of the Microsoft SQL Server Database Engine detects a deadlock, it chooses a transaction as a deadlock victim, terminates the current batch, rolls back the transaction and returns an error message to the application.

Deadlocks cannot be completely avoided; however, following certain coding conventions can minimize the occurrence of deadlocks.

Disk queue length

Disk queue length tracks the average number of I/O requests that are queued and waiting for a disk during a sampled interval. This figure may include I/O activity generated by processes other than SQL Server. Values that exceed the threshold set in this metric indicate a bottleneck.

Disk transfer time

Reading or writing data requires a disk to access the disk sector where the requested data resides. After this sector was accessed, the amount of time required for a disk to read or write data from or to a storage media is referred to as disk transfer time.

Transfer time, usually expressed in milliseconds, is part of the disk access time, that is, the total time required for the computer to process the data request from the processor and then retrieve the needed data from a storage device.

Disk utilization

The percentage of elapsed time that a disk is busy servicing I/O requests.

DiskPerf

A Windows command line utility, which enables or disables the collection of I/O statistics.

Distributing instance

A distributing instance, also known as Distributor, is a database instance that stores replication-specific data associated with one or more publishing servers.

The distributing instance includes several databases. Each of these databases is associated with a single publishing server, and stores replication status data, metadata about the publication, and, in some cases, acts as a queue for data moving from the publishing servers to the subscribing servers.

The distributing instance can take part in the replication process in either of the following methods:

- Local Distributor — when a single database server instance acts as both the publishing server and the distributing instance.
- Remote Distributor — when the publishing server and the distributing instance are configured on separate database server instances.

Distributor

See *Distributing instance*.

DMO

Distributed Management Objects. A set of COM objects is used for maintaining SQL Server objects such as databases, tables and indexes, as well as to administer SQL Server itself. Also called SQL-DMO.

Drilldown

A Foglight for SQL Server view that provides more detailed information than what is available on the main window. Often contains charts or tables showing SQL Server or Windows statistics or objects.

DTC

DTC (Distributed Transaction Coordinator) service is an integral component of Microsoft Windows (on Windows 2003 and higher versions), and is also available as part of the SQL Server installation if a previous version of Windows is used. This service is responsible for coordinating distributed transactions, that is, transactions that involve multiple resource managers, such as databases, message queues, and file systems.

DTS

DTS (Data Transformation Services) is a SQL Server component which provides a set of tools that allow extracting, transforming and consolidating data from various sources (for example, Oracle database, Excel spreadsheet or Microsoft Access) into single or multiple destinations that use uniform SQL Server format.

DTS can be customized to efficiently remap columns, convert data types, and change the database schema.

E

Error log

SQL Server error log is a standard text file that holds SQL Server information and error messages. This file can provide meaningful information for handling SQL Server-related issues. This information can either be used for tracking down problems or alert to potential or existing problems.

Event alert

A SQL Server Agent object that performs an action when a specified type of message is encountered by SQL Server. The action performed can be one of the following:

- Sending e-mail, pager, or Net Send messages
- Running an SQL Agent job

Execution contexts

Execution contexts, also known as executable plan, are created when, during the execution of a compiled plan, SQL Server keeps track of information about the state of execution and stores this information in the plan cache.

Extended stored procedures

A SQL Server object that dynamically loads and executes a function within a dynamic-link library (DLL) in a manner similar to a stored procedure. Actions outside SQL Server can be triggered, and external information returned to SQL Server. SQL Server comes pre-installed with a large number of extended stored procedures. In addition, SQL Server includes system-stored procedures that allow adding and dropping extended stored procedures, or provide information about such procedures.

Extent

The unit of space allocated to a SQL Server object, such as a table or index. In SQL Server, an extent is eight continuous pages, or 64 KB.

External procedures

An external procedure, also sometimes referred to as an external routine, is a procedure stored in a dynamic link library (DLL) on Windows platform or shared library under UNIX. The external procedure is registered with the base language, and then invoked to perform special-purpose processing.

Because external procedures run in a process separated from the database instance, using these procedures ensures that any problems on the client side do not adversely affect the database.

F

File

An on-disk structure in which SQL Server stores data and log information. Corresponds to a Windows File.

Filegroup

A filegroup is a logical storage unit that consists of one or more database objects. The filegroup, which forms a single unit of allocation and administration, maps to a file system file or multiple files.

Each database has at least one File Group (DEFAULT), which stores all system objects. When creating user-defined objects such as a table or index, it is possible to define a custom file group (usually comprising several files and disks) on which these objects reside.

File cache

An area or physical memory that Windows has allocated to speed the reading and writing of files from disk. SQL Server data and log files bypass the Windows file cache, as SQL Server already caches these files in its own memory structures.

Foglight Agent Manager

The Foglight Agent Manager manages certain types of agents installed on monitored hosts. It provides a centralized communications link between the Foglight Management Server and these agents and manages the agents' life cycles. The Foglight Agent Manager also provides a number of support services such as deployment, upgrade, and the ability to configure agents.

For further details, see *Foglight Administration and Configuration Guide*.

Foglight Management Server

The Foglight Management Server (FMS) is the central component of Foglight. The Management Server receives information from agents and makes it available in the browser interface.

The Foglight database stores all system, application, and performance data. Over time, it becomes an invaluable source of historical information for planning future system capacity requirements and for doing point-in-time analysis. The information in the database can also be made available for use in external systems.

The Management Server performs continuous internal monitoring on itself and other Foglight components, as well as on the Java Virtual Machine (JVM) and the database.

Foreign key

A column or columns within one table that relate to the primary key within a master or a parent table.

Forwarded records

When a row in a table that does not have a clustering index is updated, and if the modified row no longer fits on the old page, then SQL Server moves the row to a new page and leaves a forwarding pointer in the old page. The forwarding pointer saves the need to modify the non-clustered indexes to make them reflect the row's new location; however, extra I/O resources are now required for subsequent retrieval of this row using non-clustered indexes.

A high rate of forwarded records can indicate a need to reorganize tables (unload/reload) or define clustered indexes.

Free buffers

See Free List.

Free list

The set of buffer pages that are currently available for immediate re-use. These are SQL Server memory pages that are not currently allocated to any cache. The Free List is maintained by the Lazy Writer process (see glossary definition of Lazy writer on page 295). If the Free List becomes empty, free pages are not immediately available to SQL Server when required, and the connections needing them might have to wait while SQL Server makes buffers available. This can result in lower performance.

Free pages

Pages located in one of the SQL Server cache memory types, which are available for immediate reuse.

The Lazy Writer Process periodically scans all SQL Server caches, and maintains a list of such "free" pages.

When SQL Server needs a free memory page (for example, when reading a database page from disk into the buffer cache), and no free pages are immediately available, the connection needing the free page must wait while SQL Server makes buffers available. This results in slower performance. In the worst case, the connection will have to wait while SQL Server writes a modified page out to disk in order to make a free buffer.

Free pages values that are consistently close to zero may indicate use of inefficient queries, or shortage of SQL Server cache memory.

Free potential

The term Foglight for SQL Server uses to describe the amount of space that is available for allocation for a File Group. Free potential is defined as the amount of free space in the file group, plus the file group's Potential Growth (the amount by which the file group's files could grow).

Free space

The amount of disk space within a Database or a File Group that can be allocated to any table or index as required, but is currently allocated to no table or index. See also glossary definition of Unused space on page 307.

Freespace scans

A Freespace Scan is initiated when SQL Server needs to determine where a new record can be inserted.

Full text search

A Windows service that can optionally be installed with SQL Server. This service, which is also called MSSearch service, allows indexing selected columns in a way that allows users to search any word, phrase, or derivative of any word contained in that column.

Full Text indexes are not updated in real time.

G

GAM

Global Allocation Map. Pages in SQL Server databases that record the extents that have been allocated, and can therefore be used for locating free extents in a database.

Ghosted records

When a record is deleted from a table, to improve concurrency SQL Server does not physically remove the row from index Leaf levels. Instead, the record is marked as deleted (ghosted). At a later point, a housekeeping process asynchronously removes the ghosted records from the Leaf level. Until the records are removed, SQL Server must skip the ghosted records during Leaf level scans.

Growth increment

The amount by which a file grows when SQL Server decides to extend it. This amount can be expressed in an absolute size (megabytes — MB), or as a percentage of the current file size.

GUID

Globally Unique Identifier. A 16-byte binary value that is guaranteed to be unique. Corresponds to the SQL Server *uniqueidentifier* data type.

H

Hash buckets

SQL Server pages in memory, which contain an array of pointers to buffer pages, thereby allowing faster allocation of a page in memory. Can be thought of as conceptually similar to an index into SQL Server's buffer cache.

Hashing

In general, hashing refers to the technique of mathematically transforming a key value into a relative address that can be used to rapidly locate records.

Heap

The name given to tables that do not have a clustered index. The rows in these table are not stored in any specific order.

Hit rate

A ratio between logical accesses and physical accesses. Indicates how much work (I/O, compilations, etc) is being saved by caching information in memory.

Hit ratio

See Hit rate.

Host name

The name of the client computer that established the SQL Server connection.

Host process

A string that uniquely identifies the instance of the client application that established the SQL Server connection.

I

I/O

Input or output to a peripheral device. In a database context, I/O refers to refers to input or output to disk devices.

Index

A database object that provides fast access to data in the rows of a table, based on key values. Indexes provide quick access to data and can enforce uniqueness on the rows in a table.

Indid

A number that uniquely identifies an index on a table. All tables have at least an indid of 0 or 1. Indid 1 corresponds to a Clustered index, indid 0 corresponds to a Heap structure. Also called IndexID.

Integrated security

See Windows authentication mode on page 308.

Intent Locks

Intent locks are locks used by the database in order to protect a resource that is found lower in the lock hierarchy from a shared (S) lock or exclusive (X) lock. Because these locks are acquired before a lock at the lower level, they signal intent to place locks at a lower level.

Intent locks serve the following purposes:

- Preventing the modification of higher-level resource by other transactions, in a way that would invalidate the lock at the lower level
- Improving the Database Engine's efficiency in detecting lock conflicts at the higher level of granularity

Intent locks include intent shared (IS), intent exclusive (IX), and shared with intent exclusive (SIX).

J

Job

An SQL Server Agent object that can be scheduled to execute any number of tasks. A Job consists of one or more steps, which can each run SQL statements, Windows commands, and so on.

K

Kernel memory

The physical memory allocated to Windows kernel.

Kernel mode

See Privileged mode on page 301.

Kill

A Transact-SQL statement that terminates a SQL Server connection. Any outstanding transactions for the selected session are rolled back and all locks are released.

L

Latch

A Latch is like a mini-lock that is used internally by SQL Server to serialize access to certain resources. Latches do not participate in transactions and are typically held for very short durations.

Lazy writer

A SQL Server process that periodically scans all SQL Server caches, and maintains the Free List, which is the collection of buffer pages available for immediate re-use. If necessary, the Lazy Writer process writes modified pages out to disk in order to make them available. A high value of lazy writer writes may indicate non-optimal buffer cache memory size or use of inefficient queries.

Licensing

The legal agreement that determines how many users can access each SQL Server installation. See glossary definitions of Per seat licensing on page 299 and Per server licensing on page 300.

Lightweight pooling

A SQL Server configuration parameter that causes SQL Server to schedule its sessions as Fibres instead of Threads.

Lock

An object used by a software, such as SQL Server, to indicate that a user has an interest in a resource, and to prevent multiple users accessing the same resource in conflicting ways simultaneously. The simplest example is to prevent two people from updating the same information at the same time.

Lock area

An area of memory used by SQL Server for managing locks.

Lock escalation

The process of converting many lower level locks into fewer higher level locks, and reducing system overhead. For example, if a user has many row-level locks on a table, SQL Server may escalate these into a single table-level lock.

Lock mode

A lock mode (for example: Shared, Exclusive, Intent Exclusive) indicates the level of dependency a user has on a lock, and is used for determining which locks can be granted simultaneously on a resource.

Log

An on-disk structure, which contains a record of a database's transactions. A log consists of one or more Files.

Log cache

An area of memory used by SQL Server to speed up access to Logs. Used primarily for rollback processing.

Log writer

A SQL Server background service that writes all changes from the Log Cache out to the log files on disk. This process is termed a log flush.

Logical I/O

I/O operation performed in the memory, thereby requiring no disk access. Logical read operations use a page that was already found in memory. When logical write operation takes place, the modifications are completed in memory and only some time later written to disk.

LRU

Least Recently Used. Often refers to pages in memory. The LRU list is the mechanism by which SQL Server determines which pages should be moved to the free list first.

LSN

LSN (Log Sequence Number|), which provides a unique identification of a point in a database's log, is used for determining determine when a page was last modified.

M

Master

A SQL Server system database that contains information about SQL Server's configurations, and the location of all other databases.

MaxSize

A parameter on each SQL Server file, which determines the maximum size to which the file can grow (assuming there is sufficient free space on the disk).

MDAC

Microsoft Data Access Components. A layer of software that provides higher-level software with the ability to communicate with SQL Server. Can be downloaded for free from Microsoft's Web site.

Metric

A Unit of Measurement/Indicator that can be applied to a database. Metrics can help gauging the performance of a system.

A metric is an individual piece of information that Foglight for SQL Server has collected about the performance of a system. The information may be a numeric value (a number or percentage), a string of text, or some other valuable piece of data.

Every time that the Foglight for SQL Server window is refreshed, the cartridge retrieves the latest value of the metric, which can then be displayed in a drilldown or on the home page.

Misc.normalized trees

Normalized trees for views, rules, computed columns, and check constraints.

Miscellaneous normalized trees are used in SQL Server 2005/2008.

Model

A SQL Server system database that is used as a template when new databases are created.

Monitor page file

The monitor page file contains the Metrics and Threshold settings for a Server. This file is created when the standard settings are updated, either through calibration or configuration.

N

Named pipes

A network protocol that client applications can use for communicating with SQL Server. See Net Library.

Net library

A layer of software that enables SQL Server to communicate with client applications (and conversely). Enables communication using a variety of network protocols such as TCP/IP, Named Pipes, and IPX/SPX.

NIC

Network Interface Card. A piece of hardware that connects a computer to the network.

Null

Indicates that a value is missing, unknown, or inapplicable.

O

OBID

Object ID. A unique number that identifies a SQL Server object within a database. Objects can be tables, views, stored procedures, triggers and so on.

Object plans

Query plans generated by creating a stored procedure, function, or trigger.

OLAP

Online Analytical Processing. OLAP allows users to analyze database information (usually historical data) from multiple database systems at one time. Instead of providing data in a multi-dimensional (row and column) format, OLAP data is multidimensional, thereby facilitating the grouping, summarization and comparison of data in various ways. Such reports include, for example, comparing purchases in Q1 to purchases in Q2 to purchases of a branch in another country in Q1. These reports provide business managers with data required for trend analysis and decision making.

OLAP service

An optional part of SQL Server, which runs as a Windows service to provide OLAP facilities.

OLTP

Online Transaction Processing. OLTP allows real-time processing of SQL transactions, in order to support CRM, ERP, and other time-critical applications. OLTP is characterized by high rates of index lookups, single row modifications, and frequent commits.

Because real-time transaction processing is being increasingly carried out on a network, and can include more than one company, OLTP databases use client/Server processing and allow transactions to run on different computer platforms in a network.

Optimizer

A component of SQL Server, which takes the Query Tree and produces an optimal Execution Plan based on data sizes, indexes, join techniques and CPU, I/O & memory estimates. For further details, see Compile on page 287.

Optimizer cache

An area of memory used by SQL Server for managing the optimization of SQL statements.

osql

A command line utility that allows issuing SQL statements. Uses OLE-DB to connect to SQL Server.

P

Page life expectancy

The length of time in seconds that a database page stays in the buffer cache without being accessed, before it is flushed out.

Small values (less than 300) indicate that pages are being flushed out of the cache within a small period of time.

Paging

Disk I/O activity done by the operating system to manage its virtual memory. High paging rates can adversely affect performance.

Panel

A group of related components on the Foglight for SQL Server Instance Home page.

Parse

The process of breaking an SQL statement down into a format that is more easily understood by software. SQL statements are parsed before they are compiled into an execution plan.

Parser

The part of SQL Server that performs all parsing.

Per seat licensing

The SQL Server licensing mode, which enables deploying multiple instances of SQL Server, and allows given users to access all of these instances. See Licensing on page 295 and *Per server licensing*.

Per server licensing

The SQL Server licensing mode that allows supporting a stated maximum number of simultaneous users for a specific installation of SQL Server. See Licensing on page 295, and *Per seat licensing*.

Performance alert

An SQL Server Agent object, which performs an action when the value of a performance monitor counter satisfies a condition. The action performed can be to send e-mail, pager, or Net Send messages, or to run an SQL Agent job.

Physical I/O

An I/O operation that is performed at the hardware level. See also Logical I/O on page 296.

Physical read

A disk read I/O where the requester must wait for the disk read operation to complete. In SQL Server, the requesting session waits while the page is read from disk. This is normally the most common type of read operation in SQL Server.

Physical write

A disk write I/O where the requester must wait for the disk write operation to complete. Normally SQL Server sessions do not wait for data and index write operations to complete.

Most modifications to data and index pages are made in the Buffer Cache, and after the change has been recorded in the Log, the user can continue without having to wait for the data and index pages to be written to disk. However, certain operations, such as create index, bulk insert, and restore, can require the user to wait for the data and index writes to complete.

PID

Process ID; a unique number that identifies a Windows process at any given point in time.

Pinned

A table-level option that can be used to force pages from that table to remain in memory after they are accessed. After a page of a pinned table or index is read into the Buffer Cache, it remains there until SQL Server is restarted.

Plan

See Query plan on page 302.

Plan cache

An area of memory that SQL Server uses for storing execution plans so that they can be re-used, thereby avoiding the Recompile procedure. This area of memory was formerly known as Procedure cache. For details, see Recompile on page 303.

Potential growth

The term that Foglight for SQL Server uses to refer to the amount of disk space by which a File, File Group, Log, or Database can expand. This is a combination of the file's Auto Grow parameters, current size, and the available free disk space.

Prepared SQL plans

Query plans corresponding to statements prepared using *sp_prepare*, *sp_cursorprepare*, or using auto-parameterization.

Primary key

A column or columns that have been formally declared as uniquely identifying a row in a table.

Privileged mode

An operational state of hardware or software that has the highest priority. This mode is primarily used by operating system services such as I/O and paging. Also known as Supervisor Mode and Kernel Mode. See User mode on page 307.

Procedure cache

See Plan cache on page 300.

Procedure plans

Query plans generated by creating a stored procedure.

Process

An instance of an application executing in Windows.

Profiler

A Microsoft tool that is part of SQL Server and allows tracing events within SQL Server itself.

Publication database

A database residing on the publishing server (Publisher), which is the source of data and database objects that are to be replicated.

Publisher

See Publishing server on page 302.

Publisher databases

See Publishing server on page 302.

Publishing server

Publishing server (also known as *Publisher*) is a SQL Server instance that makes data available to other locations through replication, by hosting one or more published databases. Each of these databases can have one or more publications, which define a logically related set of objects and data to replicate.

Pull subscription

In SQL Server replication, pull subscriptions acquire the replicated data from the publishing server using periodic updates, initiated by the subscribing server. When pull subscriptions are used, each subscribing server can specify the preferred update time. Pull subscriptions are mainly used for replicating data to many subscribers, or for load balancing of a heavily loaded network.

Pulse

Pulses in the Foglight for SQL Server home page move in the direction of dataflows. As the rate of data transfer increases, so too does the speed of the pulse. Pulses can change color when the statistic represented by the pulse moves to another threshold.

Push subscription

In SQL Server replication, push subscriptions require the publishing server to push any changes made to the data out to the subscribers. Push subscriptions are mainly used when data needs to be replicated in near real time.

Q

Query plan

The method that SQL Server has determined to be the optimal way to satisfy an SQL statement. Also known as Execution Plan.

R

RAID

Redundant Array of Inexpensive Disks. RAID is used for describing the configuration of multiple physical disks into one logical disk. Windows supports both hardware RAID and Software RAID. The main RAID types are as follows:

- RAID 0 — referred to as disk striping. Each part of the logical disk is spread out over multiple physical disks. Provides very good read and write performance, but no failure recovery.
- RAID 1 — referred to as disk mirroring. Provides good recovery, but average performance.
- RAID 5 — referred to as disk striping with parity. Provides good recovery and read performance, but sub-optimal write performance.
- RAID 10 - (RAID 0 + 1) — referred to as disk striping and mirroring. Provides good recovery and read/write performance.

Random I/O

I/O in which a specific disk block is directly accessed. This is typically the I/O that results from index lookups.

Read ahead

The process by which SQL Server anticipates which pages a session will require, and reads them into the Buffer Cache before they are requested by the user connection. Used when SQL Server determines that a session is accessing table or index pages in a sequential manner. If the sequential processing does indeed continue, these pages will already be in the cache memory when required, thereby avoiding the need to wait for an I/O operation.

Recompile

The process of compiling a stored procedure part way through that procedure's execution.

Referential integrity

Referential Integrity ensures that foreign keys correctly map to primary keys. A referential constraint prevents the insertion or update of foreign keys for which there are no matching primary keys, by either preventing the deletion of primary keys if foreign keys exist, or deleting these foreign rows (DELETE CASCADE).

Relational data engine

A major functional part of SQL Server. Responsible for the parsing and optimization of SQL requests, controls query plan execution, and processes row sets from the storage engine.

Replication procedure plans

Query plans of a replication system stored procedure.

Role

An administrative group within SQL Server, which contains SQL Server logins, Windows logins, groups, or other roles. Roles can be thought of as a group of users, and can be assigned permissions.

Rollback

The process of undoing all database modifications that have been performed by a session since the beginning of its current transaction.

S

sa

System Administrator. A SQL Server login that is created by SQL Server installation. Sa is part of the SYSADMIN system role, and as such has full access within SQL Server.

Schema locks

A lock mode that is used when compiling a query. Schema locks are used for preventing changes to the underlying tables structures while the query is compiled. See Lock mode on page 296.

Sequential I/O

I/O in which disk blocks are read in sequence. This is typical of the I/O that results from full table scans.

Session

A single connection from a client application to SQL Server. Many applications have multiple SQL Server sessions open at any point in time. See SPID on page 305.

Severity

Represents how critical an alarm is. A severity determines how Foglight for SQL Server behaves when the values for a metric fall within a user-defined range of values. A severity specifies whether the information returned in the metric represents normal or abnormal behavior for the system under diagnosis. For example, unusually high values might mean that a metric has crossed a threshold into a high severity state. This, in turn, could change the color of a component on the home page; for example, from orange to red when moving from a critical to a fatal state.

The severity determines what action Foglight for SQL Server takes when a metric value falls into the range defined by a threshold.

SGAM

Shared Global Allocation Map. Pages in SQL Server databases, which record the extents that are currently used as mixed extents and have at least one unused page.

Shared locks

Shared locks are applied to enforce a read-only state on a data page, allowing other transactions to acquire a shared lock even when the first transaction's run has not been completed.

By default, SQL Server releases shared page locks after completing the page scanning is complete, rather then holding these locks until the statement completes or until the end of its transaction.

Show advanced options

A SQL Server configuration option that turns on the display of advanced configurations.

SMP

Symmetric Multi Processing. An SMP computer contains multiple equivalent CPU units.

Sort, Hash, Index Area

An area of memory used by SQL Server to manage sorting and index creation.

SPID

A unique number that identifies each SQL Server session at a given point in time. See Session on page 304.

Spike

An abnormally high value in a dataflow or graph.

Spinner

In the Foglight for SQL Server home page, a Spinner displays the current level of activity for a statistic that is not directional (for example, CPU usage). As the load increases, so too does the speed of the spin. If the statistic represented by the flow moves to another threshold, the spinner may change color. The combination of movement and color makes it easy to spot congested areas.

SQL Agent Mail

A MAPI session created by the SQL Server Agent service in order to send e-mail and pager messages.

SQL Mail

A MAPI session created by the MS SQL Server service. This allows SQL Sessions to send e-mail and pager messages using extended stored procedures.

SQL Plans

Query plans corresponding to statements prepared using sp_prepare, sp_cursorprepare, or using auto-parameterization. This definition includes also user-parameterized queries.

SQL Server Agent

A part of SQL Server that schedules and controls the execution of jobs, and monitors SQL Server for any defined alerts. Runs as a Windows service.

SQL Server authentication

The authentication mode where the user passes a SQL Server login name and password, which is checked against a SQL Server system table. The other authentication mode used is Windows authentication. For additional details, see Windows authentication mode on page 308.

SQL Server books online

See Books Online on page 284.

Standard deviation

A statistical function of SQL Server, which measures how widely values deviate from the mean (that is, the average).

Stolen pages

Pages “stolen” from the buffer pool to be used for miscellaneous server purposes, when Windows requires memory for another application. A large number of stolen pages may be normal for a short period (for example: if the system backup begins after large database batch run completes). However, constantly high value of this metric may indicate an overall system memory shortage.

Stored procedure

A series of Transact-SQL statements, stored as an object within a SQL Server database, that can be called (executed) by SQL Server sessions with appropriate permissions.

Support service

The term Foglight for SQL Server uses to identify SQL Server-related features that can be installed, and stopped or started independently from SQL Server. These services often run as Windows services in their own right. Examples are SQLServerAgent, MSDTC, and SQL Mail.

SYSADMIN role

The Server role that has full access to all SQL Server facilities. By default, this role contains the sa login and the Windows “Administrators” group. The use of Foglight for SQL Server is limited to members of the SYSADMIN role.

T

TDS

Tabular Data Stream. A private protocol self-describing data stream, which SQL Server uses in the “conversation” between client and Server.

TempDB

A SQL Server system database used for temporary storage and work areas. This database, which stores all temporary tables and stored procedures, is used by SQL Server for large sort operations.

Temporary tables and table variables

Temporary tables are tables created in the *TempDB* database. These tables are session-specific tables, that is, even though they are backed by physical disk and logged into the transaction log, the tables are automatically dropped when the session is closed. As a result, temporary tables are usually not written on disk, but only saved in the plan cache memory.

Table variables, on the other hand, are created in the memory and exist there until the running of a single Transact-SQL (T-SQL) batch is completed. Unlike temporary tables, they do not require locking and are less I/O-intensive.

Threshold

A range of values that may be returned by a metric. If the metric falls within this range, Foglight for SQL Server checks the threshold's severity to determine how to behave. For example, the component representing this metric might change color.

Torn page detection

A database-level option that causes SQL Server to detect I/O operations that were partially complete due to system crashes (for example, power failures).

Transaction

A group of one or more database modification statements, which are combined into a logical unit of work that is either wholly committed or rolled back.

Trigger

A series of Transact-SQL statements, which is stored as an object within a SQL Server database and linked to a data modification statement and a specific table. Whenever data in that table is modified, the trigger is executed by SQL Server, and can perform complex data validation or extra processing.

Trigger plans

Query plans generated by creating a trigger.

Truncate

Transact-SQL statement that removes all data from a SQL Server table. Truncate is much quicker than a Delete statement for deleting all data from a table.

Trusted

See Windows authentication mode on page 308.

U

UMS

User Mode Scheduler. The mechanism used by SQL Server to implement lightweight pooling.

Unused space

Disk space within a Database or File Group that is allocated to a table or index, but currently does not have any information stored in it. It is free space that can be used only by the table or index to which it is allocated. See also glossary definitions of Free space on page 292 and Extent on page 291.

User connection area

An area of memory used by SQL Server to manage session-specific data.

User mode

A processor mode that is primarily used for executing application code. Unlike the Privileged (Kernel) Mode, processes running in the user mode do not interact directly with the operating system's kernel, memory, or

hardware. As a result, whereas a crash of a kernel mode process in most cases leads to the entire system's crash, a crash of a user mode process usually only terminates the application that was run using this mode. See Privileged mode on page 301.

V

Virtual log file

All SQL Server logs are internally broken down into smaller units called Virtual Log Files (VLFs). The virtual log file is the unit of allocation for logs.

VLF

See Virtual Log File.

W

Waitfor

A wait type which determines that the execution of a batch, stored procedure, or transaction is blocked until a specified time or time interval is reached, or a specified statement modifies or returns at least one row.

Windows authentication mode

The authentication mode where SQL Server uses Windows facilities to determine who the connecting user is. The user or application does not need to supply a user id and password to access SQL Server because they have already logged in to Windows. See Authentication on page 283. The other authentication mode used is SQL Server authentication. For additional details, see SQL Server authentication on page 305.

Working set

The set of pages, in the virtual address space of the process, which currently reside in physical memory.

When a process references a page that is not part of its working set, a page fault occurs. The system page fault handler attempts to resolve the page fault and, if it succeeds, the page is added to the working set.

A process has an associated minimum working set size and maximum working set size. If not enough memory is available, memory requests from a new process may lead the Windows memory manager to attempt to accommodate the new allocation request by trimming the working set of other processes. Such a scenario may increase the frequency of page faults.

About us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit https://www.quest.com/.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.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. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.