

Foglight® for Azure SQL Database 6.1.0.10

Monitoring Azure SQL Database Systems

User and Reference Guide



© 2022 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.

Contents

Installing and Configuring Agents	7
Installation	7
Installing and Configuring Agents	7
Upgrading to the Current Version	7
About Monitoring Extensions	8
SQL Performance Investigator Extension	8
Installing and Monitoring a Single Azure SQL Database	8
Using Foglight <i>for Azure SQL Database</i>	10
Viewing the Databases Dashboard	10
Selecting an Instance to Monitor	11
Filtering the Display by Severity	12
Creating User-defined Database Groups	12
Components Shared by All Foglight <i>for Azure SQL Database</i> Screens	13
Using the Currently Selected Database Group Table	14
Foglight <i>for Azure SQL Database</i> Overview Dashboard	14
Breakdown and Baseline Chart Formats	1
5	
Home Page Toolbar	15
Overview view	15
SQL Performance Investigator (SQL PI)	16
Performance Tree	16
Viewing Historical Metrics	17
Blocking History	17
Viewing Execution Plans	18
Azure SQL Activity Drilldown	18
Viewing the Blocking (Current) panel	18
Viewing Detailed Sessions Data	20
Administering Foglight <i>for Azure SQL Database</i>	22
Opening the Databases Administration Dashboard	22
Reviewing the Administration Settings	22
Customizing Alarms for Foglight <i>for Azure SQL Database</i> Rules	23
Introducing the Alarms View	23
Modifying Alarm Templates	23
Configuring Email Notifications	25
Cloning Agent Settings	29
Reviewing Rule Definitions	29
Defining Connection Details	30
Defining the Connection Settings for the Monitored Azure SQL databases	30
Validating Connectivity and Starting to Monitor the databases	30
Administering SQL Performance Investigator	31
Reporting	32

Generating Reports for a Foglight <i>for Azure SQL Database</i>	32
Reference	33
Overview of Collections	33
Usability	33
Query Store	34
Query Store Top SQLs	35
Version Info	36
Resource Stat	37
Wait Statistics	38
Database Connections Summary	38
Database Replication	39
Database Storage	40
Database Recommendations	41
SQL Performance Investigator Metrics	42
Active Connections	42
Active Time	42
Active Time per Execution	43
Active Time Rate	43
Always On	43
Average CPU Percent	43
Average DTU Percent	43
Average Memory Usage Percent	43
Average SQL Response Time	43
Average XTP Storage Percent	43
Average Data IO Percent	43
Average Log Write Percent	43
Backup Recovery	44
Blocked Sessions	44
Connections	44
CPU Limit	44
CLR Wait	44
CLR Wait per Execution	44
CLR Wait Rate	44
CPU Time	44
CPU Time per Execution	44
CPU Time Rate	44
CPU Wait	45
CPU Wait per Execution	45
CPU Wait Rate	45
Cursor Synchronization	45
Database Replication	45
Deferred Task Worker	45
Distributed Transaction	45
DTU Limit	45
Elapsed Time	45
Executions	45
External Procedure	46

Full Text Search	46
Governor Wait	46
Governor Wait per Execution	46
Governor Wait Rate	46
Granted Memory	46
Hosted Component	46
Inactive User Connections	46
I/O Bulk Load	46
I/O Completion	46
I/O Data Page	47
I/O Wait	47
I/O Wait per Execution	47
I/O Wait Rate	47
Idle Time	47
Internal Cache Latch	47
Latch Buffer	47
Latch Savepoint	47
Latch Wait	47
Latch Wait per Execution	47
Latch Wait Rate	48
Lead Blockers	48
Lock Bulk Update	48
Lock Exclusive	48
Lock Intent	48
Lock Schema	48
Lock Shared	48
Lock Update	48
Lock Wait	48
Lock Wait Rate	48
Lock Wait per Execution	49
Log Buffer	49
Log Other	49
Log Synchronization	49
Log Wait	49
Log Wait per Execution	49
Log Wait Rate	49
Log Write	49
Logical Reads	49
Logical Reads per Execution	49
Max Degree of parallelism	50
Max DTU Percent	50
Max Session Percent	50
Max Workers Percent	50
Memory Wait	50
Memory Wait per Execution	50
Memory Wait Rate	50
Network HTTP	50
Network I/O	50

Network IPC	50
Network Mirror	51
Network Wait	51
Network Wait per Execution	51
Network Wait Rate	51
OLEDB Provider Full Text	51
Other Miscellaneous	51
Other Wait	51
Other Wait per Execution	51
Other Wait Rate	51
Parallelism Wait	51
Percent of Total	52
Plan Recompilations	52
Physical Reads	52
Physical reads per Execution	52
Preemptive Wait	52
Preemptive wait per Execution	52
Preemptive Wait Rate	52
Remote Provider Wait	52
Remote Provider wait per Execution	52
Remote Provider Wait Rate	52
Row count	53
Service Broker	53
Synchronous Task	53
Wait Time Percent	53
Writes	53
Writes per Execution	53
XTP Log write Wait	53
XTP Miscellaneous Wait	53
XTP Procedure Wait	53
XTP Transaction Wait	53
XTP Wait	54
XTP Wait per Execution	54
XTP Wait Rate	54
Glossary	55
About us	62
Technical support resources	62

Installing and Configuring Agents

This guide provides configuration instructions, conceptual information and instructions on how to use the Foglight *for Azure SQL Database* 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, as well as the cartridge's interaction with and support of additional services and modules, such as replication and virtualization.

This guide is intended for Azure SQL Database administrators and for any users who want to know more about monitoring Azure SQL Databases through Foglight *for Azure SQL Database*, and the steps required for discovering and configuring the Azure SQL agent. It is also meant for those users who want to learn about the methods used for configuring and applying user-defined settings.

This section provides information about installing the cartridge and configuring the agents for monitoring Azure SQL Database systems:

- [Installation](#)
- [About Monitoring Extensions](#)
- [Installing and Monitoring a Single Azure SQL Database](#)

Installation

For installation pre-requisites, permissions, and information necessary to determine your environment's hardware requirements, see the *Foglight for Databases Deployment Guide*.

Installing and Configuring Agents

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

Upgrading to the Current Version

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

To upgrade the cartridge to the latest version:

- 1 Deactivate the previous Azure SQL agent.
- 2 Navigate to the **Cartridge Inventory** dashboard and install the cartridge file, *DB_Azure-6_1_0_10.car*.
- 3 Navigate to the **Agent Status** dashboard and deploy the Azure SQL agent package to the existing Foglight Agent Manager hosts.
- 4 Activate the Azure SQL agent.

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 database and the environment it is running on, creating a whole and unified status.

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 Azure SQL 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

i | **NOTE:** SQL Performance Investigator requires a license. If you are using a trial version and would like to request pricing, contact <https://www.quest.com/register/57891>.

i | **NOTE:** SQL PI requires a repository database that is installed automatically on the Agent Manager.

Installing and Monitoring a Single Azure SQL Database

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

Foglight *for Azure SQL Database* 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.

i | **IMPORTANT:** When running Foglight *for Azure SQL* 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 database installation wizard:

- 1 On the navigation panel, click **Homes > Databases**.
- 2 Click **Monitor > Azure SQL** in the upper left corner of the Databases View.

The *Monitor Azure SQL Database* dialog box appears.

i | **NOTE:** If a user-defined database group is currently 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 will be running. The default is the agent manager with the least agents installed.

- a Click the **Agent Manager Host** <agent_manager> link located at the top of the Azure Database Connection section.

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.

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

- 4 Specify the name of Azure SQL Database to be monitored in the Connection Details section.
 - a **Server Name** — Specify the host name.
 - b **Database Name** — Specify SQL database name.
 - c **Port** (optional) — It is required only when the TCP/IP connection port of the Azure SQL Database is other than the default port (1433). If it is not specified, the default port (1433) will be used.
- 5 Select between *SQL Server Authentication* and *Azure Active Directory (AD) Authentication*. Specify the username and password to be used for monitoring the Azure SQL Database in the Login Credentials section.
- 6 Use the automatically generated agent name by selecting the checkbox in Configuration section. Or specify a unique agent name by deselecting the checkbox.
- 7 Optional — In the Monitoring Extensions pane, click the SQL PI monitoring extension. You are prompted to choose the Agent Manager on which the SQL PI repository will be installed.

i | **NOTE:** The SQL Performance investigator requires an additional license, check the licensing information via **Click for Licensing Information**.

- 8 **Optional** To enable SQL Performance Investigator, select **SQL PI** in the **Monitoring Extension** section. The SQL PI Repository dialog is displayed. Select the Agent Manager on which the SQL PI repository should be installed. Then click **Apply**.
 - b If the SQL PI repository existed on the selected Agent Manager, no more action is required for the configuration.
 - c If no SQL PI repository exists on the selected Agent Manager, the **SQL PI Repository Settings**

i | **NOTE:** To use gMSA authentication for connections to the PI Repository, select **Local User** in the Authentication drop-down. This requires that your FMS and Agent Manager are configured to run as a Windows Service logged on with the gMSA account. The gMSA account requires access to the SQL instance.

- d If the given SQL PI repository user does not have sufficient privileges, the **Insufficient Privileges** dialog will be displayed. Enter an admin user and password, and then click **Grant Privileges**.

- 9 Click **Monitor**.

i | **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, insufficient privileges, incorrect credentials, or an Agent Manager that reached its full monitoring capacity.

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

Using Foglight *for Azure SQL Database*

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

This section provides agent configuration instructions and information on investigating Azure SQL Database performance. It covers the entire monitoring process, starting with the discovery of the databases and the connection with these databases, and proceeding with the use of the various drilldowns and the Global Administration options.

Viewing the Databases Dashboard

i | **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](#).

The Instance home page is launched 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 of 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, Azure SQL, Oracle, SAP ASE, DB2, MongoDB, Cassandra, or All Instances) and displays the number of instances for each database type, along with a breakdown according to the instance health state severity (Normal, Warning, Critical, Fatal).

When selecting the Azure SQL Tile, an Azure SQL Databases table will be displayed. It shows the “regular” Azure SQL Databases.

- 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 of the currently monitored instances for each database type, divided by their health state

(for example: four Azure SQL Databases, 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.

i | **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 12.

- The **Monitor** button — Use this button to add instances to be monitored.
- 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 — 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, as well as configuring the connection to SQL Performance Investigator (if installed). For details, see [Administering Foglight for Azure SQL Database](#) on page 22.
- **Currently selected database group table** — a list of all monitored databases within the database group that is currently selected in the Databases section. For details, see [Using the Currently Selected Database Group Table](#) on page 14.
- The **Show Quick View** () button — provides the ability to open a quick view of the instance at the bottom of the screen.
- The **Select dashboards** () button — provides direct link to several drilldowns and panels, thereby saving the need to navigate to these locations through the Overview drilldown.

i | **IMPORTANT:** If the display is filtered to show only Azure SQL Databases the Databases table is renamed to Azure SQL.

Selecting an Instance to Monitor

Because the Databases dashboard displays by default all of the currently monitored databases, it is advisable to 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, Azure SQL).
 - **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 12.
 - **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 User-defined Database Groups](#) on page 12.
- 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.
- 4 Click **Home Page** on the cue card to launch a full-screen view of the requested instance.

i | **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 Azure SQL Database* mode.

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:

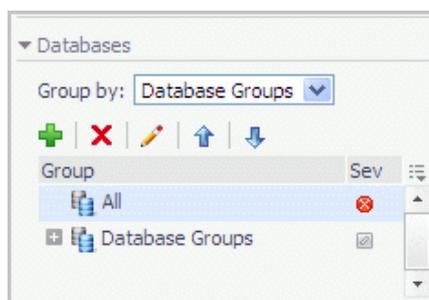
- 1 Click a certain type of severity within a specific database type (for example: Azure SQL databases whose severity level is Critical).
The selected filter is shown in the table's title.
- 2 To select another filter, click the required status indicator (for example: **Azure SQL > Normal severity**).

Creating User-defined Database Groups

Use the Databases area of the navigation pane to create, edit, and remove groups of database instances. Click **Edit Group** to add or remove database instances from the sub-group.

To create a database group:

- 1 Select the parent database instance group, *Database Group*, under Databases in the navigation pane.



- 2 Click the **+** button.
The Add Group dialog box opens.
- 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 of the databases to the Selected column.
- 6 Click **OK**.
The group name appears in the Database Groups list.
The database instances are listed in the Databases pane.

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

To add sub-groups to a selected group:

- 1 Position the cursor on the requested group's name.
- 2 Repeat [Step 2 to Step 6](#) described in [To create a database group: on page 12](#).
i | NOTE: The Add Group dialog box displays the name of the parent group in the Parent field. Before proceeding, ensure that the sub-group is added to the requested group.

To remove a database group:

- 1 Select the group to be removed.
- 2 Click the **X** button.

The Remove Groups dialog box appears, asking approval to remove the selected group with all its sub groups.

- 3 Click **Remove**.

The group is removed from the Database Groups list.

To edit a database sub-group:

- 1 Select the requested group.

- 2 Click the  button.

The Edit Group dialog box opens.

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

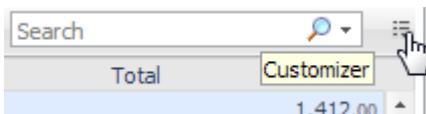
- 5 Click **OK**.

Components Shared by All Foglight *for Azure SQL Database* Screens

The tables' **Customizer** button and the **In-context action** buttons are common to all Foglight *for Azure SQL Database* screens.

The **Customizer** button — use this button, which is found on the upper right corner of each table, to create a custom filter for the table. The filter is created by clicking **Customizer** and specifying the criteria that the various values should meet in order to be displayed in the table (having a specific name, exceeding a certain size, and so on). For detailed instructions and examples, see *Foglight User Guide > Working with Dashboards > Working with Tables*.

Figure 1. Customizer button



In-context actions buttons — found on the upper right corner of all screens:

- **Refresh** — clicking this button retrieves the data acquired using the most recent sampling.
- **Agent Settings** — when using panels whose configuration is set using the Databases Administration dashboard, clicking this button opens the relevant screen in the Databases Administration dashboard (for example, clicking this button from within the **Databases > Locks** panel opens the Locks screen).

Using the Currently Selected Database Group Table

The currently selected database group table displays the following columns.

Table 1. Currently Selected Database Group Table

Name	Description
Database	
Sev	The highest severity alarm of the Azure SQL Database, which determines the database's health state.
Name	The name of the Azure SQL Database. In addition, this row includes the Go to Home Page button.
Version	The version number of the Azure SQL Database.
Up Since	The date and time that the Azure SQL Database was last restarted.
Workload	The workload (average active sessions) for the Azure SQL Database. When holding the cursor over the workload graph, the dwell displays a chart that shows the workload history over the specified time range.
DB Alarms	The number of warning, critical, and fatal alarms for the Azure SQL Database. When holding the cursor over one of the alarm counts, the dwell displays the most recent alarms invoked for 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.
System Utilization	
Host	The name of the computer that is hosting the database.
Monitoring Status	
Agent	The operational status of the monitoring agent. When the agent instance is running, the State icon () is green, and holding the cursor over the icon displays the message Collecting Data. When the agent instance is running but not collecting data, holding the cursor over the icon displays one of the following status messages listed below. <ul style="list-style-type: none">StartingStoppedStoppingUnknown When SQL PI is configured the SQL PI icon is displayed.

Foglight *for Azure SQL Database* Overview Dashboard

The Foglight *for Azure SQL Database* Overview dashboard provides various detailed views that are accessible using the toolbar.

The Overview dashboard contents depends on whether an Azure SQL Database is selected.

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 into various components. For example, space utilization is broken down by the various components that occupy the space.

Home Page Toolbar

The Foglight *for Azure SQL Database* Overview dashboard provides the following toolbar buttons.

Table 2. Foglight *for Azure SQL Database* Home Page Toolbar

Name	Description
Overview	Opens the Overview view .
SQL PI	Opens the SQL Performance Investigator (SQL PI) drilldown, which displays performance-related data at the following levels: <ul style="list-style-type: none">• High-level data — available for all instances.• In-depth data, delivered by means of the Cluster view or Instance view tree — available only for instances or RACs that have Performance Analysis installed and configured.
Activity	Opens the Viewing the Blocking (Current) panel , which contains the following panels: <ul style="list-style-type: none">• Blocking (Current)— see For more information, see Viewing the Blocking (Current) panel on page 18.• Sessions — see Viewing Detailed Sessions Data on page 20.

Overview view

The upper section of the view includes general information on the monitored database:

- Azure SQL Database
- Collation
- Elastic pool
- Edition

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 — Shows information and alarms about the availability of the database. Also includes the monitoring state for that database.
- HA/DR — High Availability (HA) and Disaster Recovery (DR) state of the database. Displays the state of the resources used for availability. For example, Always On, Cluster, Replication, Mirroring, Log Shipping.
- Storage — Shows the space utilization level for all Data and Query store.
- Compute Utilization— Provides general information collected from the server:
 - DTU— The DTU tile exists only when the database purchase model is DTU. A database transaction unit (DTU) purchasing model represents a blended measure of CPU, memory, reads, and writes.
 - CPU— Azure SQL CPU utilization compared to the general host CPU utilization.

- Data I/O— Utilization of data input or output on the monitored server.
- Log I/O— Utilization of log input or output on the monitored server.
- Memory— Amount of memory Azure SQL uses compared to other processes and the total memory on the server.
- Database Performance — Summarizes general information collected from the SQL PI dashboard. The pie chart displays the average workload divided by resources over time. The graphs display the total workload trend over time.

The Database Performance tile also displays the following issues:

- Tuning recommendations
- Current sessions state (for example, Active, Inactive, and Blocked)

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

- Alarms

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

- Display all active alarms for the underlined database.
- Enable sorting by creation time or severity.

- Top 10 SQLs — Displays the 10 SQLs with the longest duration.

SQL Performance Investigator (SQL PI)

The SQL PI view provides the ability to perform a more in-depth analysis and investigation of the database activity and resource consumption by adding a dimension view of the activity, locked objects, and blocking history.

Performance Tree

The performance tree provides iterative (up to three levels) access to any of the key dimensions associated with Azure SQL Database activity, based on the OLAP multidimensional model and an database view of the database activity. Domain nodes offer a hierarchical view of all types of Azure SQL Database 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 User, follow the steps described below:

- 1 Select the 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 Azure SQL Database dimension of interest, to gain a complete understanding of the causes of its behavior.

The default Azure SQL Database dimensions are as follows:

- SQL Statements — The executed SQL queries.
- TSQL Batches— A batch of SQL statements. That is a group of two or more SQL statements or a single SQL statement that has the same effect as a group of two or more SQL statements.
- Programs — names of the programs that connected to Azure SQL Database and executed the SQL statements.
- Users — Azure SQL Database login names used for logging in to Azure SQL Database.
- Client Machines — The machines on which the client executable (connected to Azure SQL Database) is running.
- Context Info — Displays the information set by the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure.
- Command Types — Executed SQL command type (for example, INSERT and SELECT).
- 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.

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 database resource activity over the selected time frame by emphasizing the resources by colors.
 - Baseline chart — Displays the database workload compared to the baseline over time.
 - Breakdown chart — Activity of the database by second.
 - Resource Breakdown Pie chart — Displays the resource breakdown usage by % of the total database activity.
 - All wait events pop up — Displays details of the wait events that the database 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 database 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 Database 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.

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
- SPID

- Blocked By
- Resource
- Lock Resource Type
- Status
- Duration
- Program
- User
- SQL Text
- Machine
- Command Type

Viewing Execution Plans

This view presents the execution plan of a selected SQL and the cost of it. A Historical execution plan can be generated from History by selecting the statement or batch and by clicking **Open Plan in SSMS** in the top of the table.

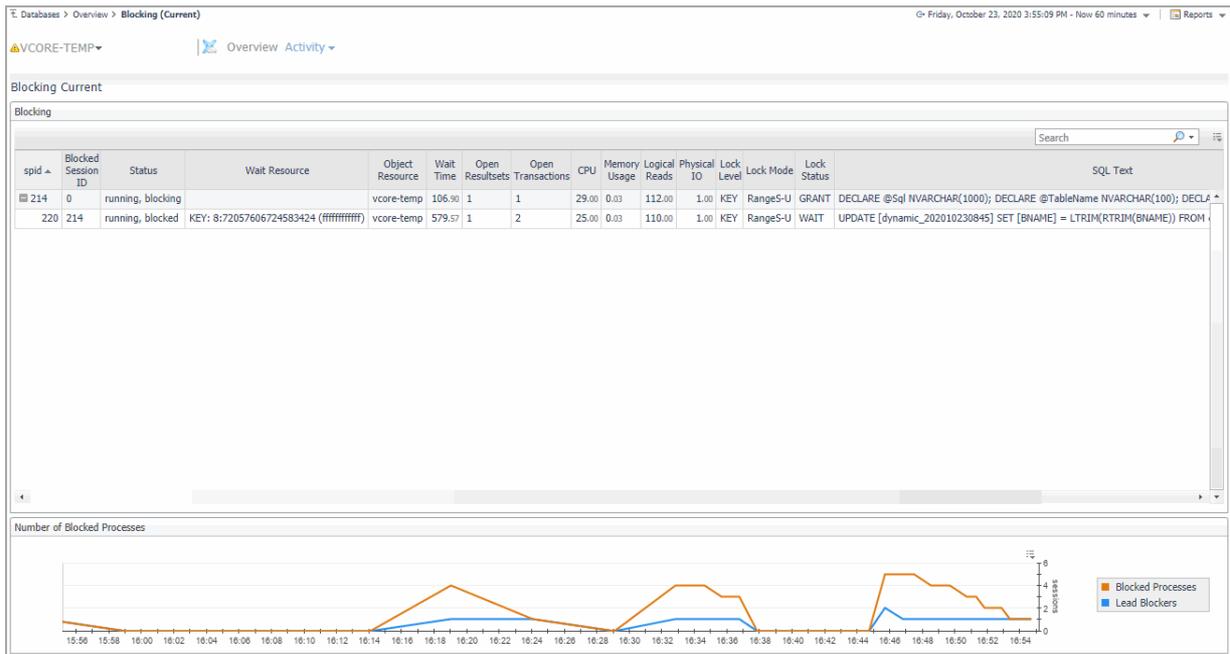
Azure SQL Activity Drilldown

Use the Azure SQL Activity drilldown to carry out the operations described in the following topics:

- Viewing information about current blocking processes — using the Blocking (Current) panel. For details, see [Viewing the Blocking \(Current\) panel](#).
- Viewing information about sessions that experienced the highest contention of a specified contention criterion — using the Sessions panel. For details, see [Viewing Detailed Sessions Data](#) on page 20.

Viewing the Blocking (Current) panel

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 20.

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 Azure SQL Database Screens](#) on page 13.

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 3. The *Blocking* table displays the following parameters:

Parameter	Description
SPID	Unique number the Server has assigned for identifying the selected session.
Blocked Session ID	Session ID that is currently blocked.
Status	Status of the session (Blocked, Blocking, or both). For sessions at the head of the blocking chain (those that are not blocked), this will indicate if the session is Runnable or Sleeping.
Wait Resource	Resource for which the session is waiting.
Object Resource	Object that is owned by the selected session and waited by the other session involved in the lock.
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.

Table 3. The *Blocking* table displays the following parameters:

Parameter	Description
Open Result sets	Number of result sets that are open for the session.
Open Transaction	Number of transactions that are open for the session.
CPU	Total amount of CPU consumed by the session so far.
Memory Usage	Total amount of memory consumed by the session so far.
Logical Reads	Number of logical reads that have been performed by the session.
Physical I/O	Total amount of I/O resources consumed by the session so far. This information can be useful when deciding which sessions to kill.
Lock Level	Locked resource type of the session.
Lock Mode	Mode of the locked session.
Lock Status	Current status of the locked session.
SQL Text	Text of the locked SQL query.

Monitoring blocked processes for the sampled interval

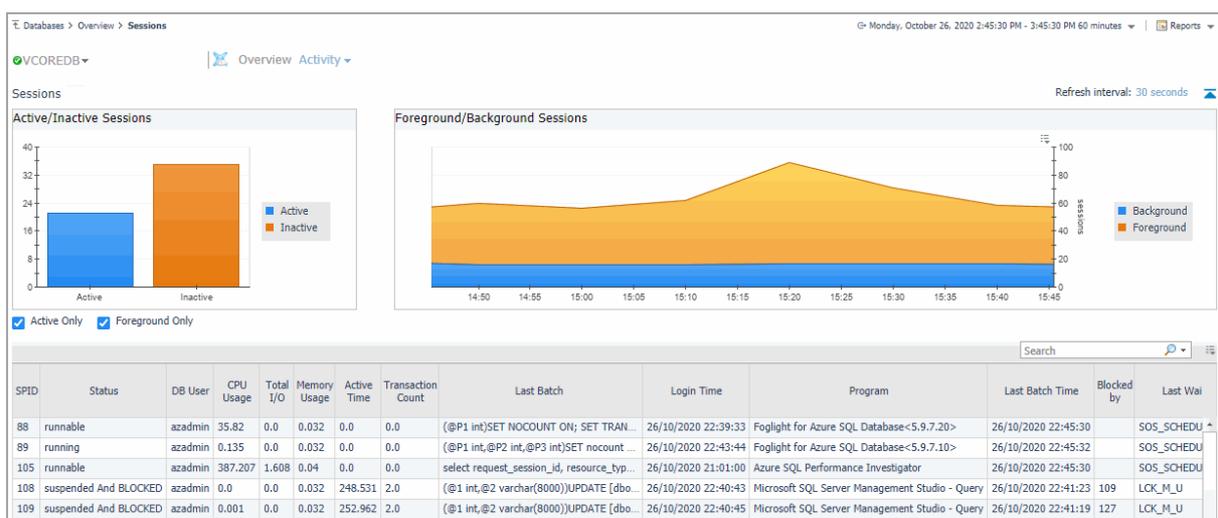
The *Number of Blocked Processes* chart displays the number of Azure SQL Database sessions that were involved in blocks over time. Use this chart to review the frequency and duration of lock conflicts in Azure SQL Database.

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).

Viewing Detailed Sessions Data

The *Sessions* screen allows you to trace the activity of all currently connected sessions, as well as their resource consumption.



NOTE: 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.

This panel can also be used to carry out the tasks described in the following sections:

Active/Inactive Sessions

The diagram displays the distribution of the currently connected sessions between active and inactive sessions.

Foreground/Background Sessions

The diagram displays the distribution of the currently connected sessions between foreground and background sessions.

Sessions List

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 user sessions, by clearing the check box, the background sessions will be presented as well.

The Sessions list provides detailed information about the sessions as follows:

- SPID — Session identifier.
- Status — Indicates the status of the session.
- DB User — The login name for this session.
- CPU Usage — The CPU usage of this session.
- Total I/O — The total I/O of this session.
- Memory Usage — The memory usage of this session.
- Active Time — The active time of this session.
- 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.
- Last Batch — The latest batch ran by this process.
- Login Time — The time when the user logged into the session.
- Program — The name of the program the user is running to access Azure SQL Database.
- Last Batch Time — The time when the last batch started execution.
- Blocked by — The session ID that blocks this session.
- Last Wait Type — The type of the last wait.
- Last Wait Resource — The resource for which the session is currently waiting.
- Physical Write — 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.
- Last Tran. Start Time — The start time of the last transaction.

Administering Foglight *for Azure SQL Database*

You use the Databases Administration dashboard to set options for collecting, storing, and displaying data about monitored Azure SQL databases.

Opening the Databases Administration Dashboard

You can edit settings for one or more Azure SQL databases on the **Databases > Administration** dashboard.

i | **NOTE:** If you attempt to select instances of more than one type of database, such as a SQL Server database and an Azure SQL Database, an error message is displayed.

To open the Databases Administration dashboard:

- 1 In the navigation panel, under **Homes**, click **Databases**.
- 2 Select the row check boxes beside one or more Azure SQL databases.
- 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 an Azure SQL tree.

i | **TIP:** The list of agents you selected can be found by clicking **Selected Agents**.

Reviewing the Administration Settings

Use the Databases Administration dashboard to set options for collecting, storing, and displaying data, which apply to all of 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 on the Databases Administration dashboard apply to all of the agents that were selected before opening the Administration dashboard. As a result, the same unit of measure and aggregation value for display are enforced for all currently selected agents.

To view the full list of selected agents, click **Selected Agents** button at the upper right corner of the screen. To change the list of agents to which the metrics apply, exit the Databases Administration dashboard, select the required agents, and re-open the dashboard.

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 of settings from the left-hand panel.
- 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.

Customizing Alarms for Foglight *for Azure SQL Database* Rules

i | **NOTE:** Foglight 6.1.0 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 Azure SQL Database* multiple-severity 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](#)
- [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 22.
- 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:
 - [Modifying Alarm Templates](#)
 - [Configuring Email Notifications](#)
 - [Cloning Agent Settings](#)

Modifying Alarm Templates

Foglight 6.1.0 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 Azure SQL Database* rules are 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 6.1.0 User Guide* for more information.

- IMPORTANT:** Avoid editing Foglight *for Azure SQL Database* 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.

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. When 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:

- [Modifying alarm threshold values](#)
- [Modifying alarm threshold values](#)
- [Configuring Email Notifications](#)

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 29.

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 6.1.0 User Guide* for more information on editing alarm templates.

Modifying alarm 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. 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.

To change severity levels and thresholds:

i | **IMPORTANT:** The procedure below is a summary. Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.1.0 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 6.1.0 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](#) on page 26.

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 4. 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 5. Configure email notification settings

Scope	Procedure
All alarms	Click All Alarms . In the All Alarms 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 Email Notification Settings tab. <ul style="list-style-type: none">• 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.

i | **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 25.

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 28.

- **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.

- 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).
i | **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 26.

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](#) on page 28, the registry variable is already open for editing in the Edit Registry Variable dashboard.
i | **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 receive email notifications based on the schedule.
i | **TIP:** The addresses may be 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.

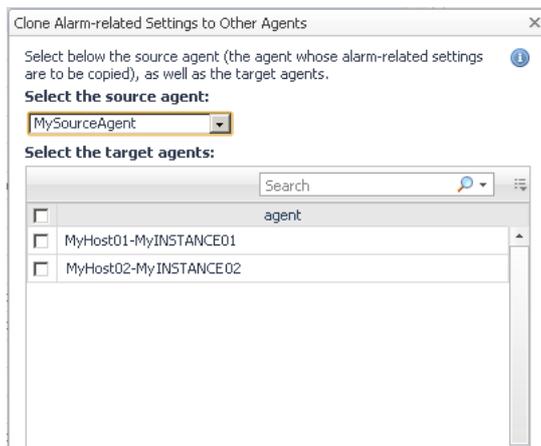
This procedure walks you through selecting the source agent from the Databases dashboard. However, you can also open the Administration dashboard with multiple agents selected. In this case, you select the source agent in Clone Alarm-related Settings to Other Agents dialog box.

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 Azure SQL Database* 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, it is recommended to copy the rule and create 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, refer to 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 Rules & Notifications dashboard, click **Rules**.
- 3 Type **Azure** in the Search field to see the list of predefined rules for Azure SQL databases.
The Foglight *for Azure SQL Database* 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 breadcrumbs to return to the dialog box.

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 enable user-defined collections and set VMware connection details.

NOTE: The following sections instructs how to define the connection settings for monitored Azure SQL databases.

Defining the Connection Settings for the Monitored Azure SQL databases

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. To cancel the selection, click **Select None** and select again.
- 2 Click **Set Credentials**.
The Edit Instance Credentials dialog box opens.
- 3 Use the Connection Details section to enter or accept the default port (1433).
- 4 Use the Specify Login Credentials section to select the type of authentication, specify the user name and the password used for connecting to the Azure SQL Database.
- 5 Click **Set** to proceed to the next stage of validating the database's connectivity.

Validating Connectivity and Starting to Monitor the databases

After setting the default credentials for the host, these newly created credentials can now be used by the wizard to attempt to log in to the databases.

To validate the databases' connectivity:

- 1 Click **Test connection**.

The Verifying Connectivity progress bar appears.

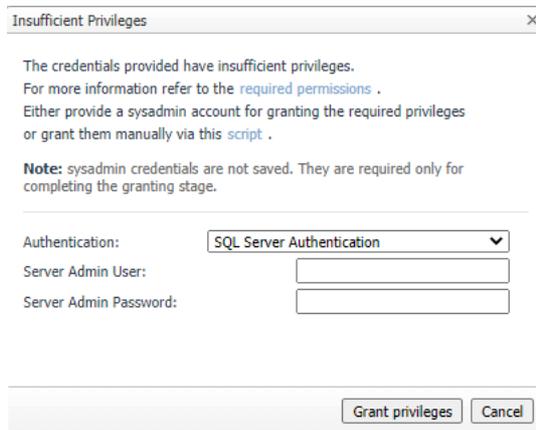
At the end of this process, any connectivity issues are listed in the Status column of the database table. When the connection is successful, the Status column displays the status message *Validated*, which indicates that the database connected successfully and the specified Azure SQL user has the required permissions.

If the connection failed verification, the Status column displays one of several connectivity status messages.

The messages, causes, and appropriate responses are:

- Invalid username/password — check the credentials and try again.
- Incorrect port— check if the SQL Azure port is correct and try again.
- Wrong Database Credentials — modify the login credentials.
- Insufficient Privileges — grant the user the privileges required for connecting to the database, by clicking the status **Insufficient Privileges**.

The Insufficient Privileges dialog box opens.



Use this dialog box to specify a Server Admin user with sufficient privileges.

Select the type of authentication. Type a Server Admin user and password, and then click **Grant privileges**.

If the Server Admin credentials entered were incorrect, the column displays the status *Wrong Sysadmin credentials*.

After correcting the mistakes that resulted in the connectivity failure, click again **Validate connectivity**.

- 2 Click **Validate connectivity** on the status bar.
- 3 Click **Save Changes**.

The Applying Modified Settings progress bar appears.

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.

Reporting

Foglight *for Azure SQL Database* enables generating reports about various aspects of the selected database. This section provides information on how to generate the various reports, as well as a brief description of each report.

i | **NOTE:** For detailed information regarding the use and configuration of reports, see *Foglight User Guide* > Working with Reports.

For details, see these topics:

- [Generating Reports for a Foglight for Azure SQL Database](#)

Generating Reports for a Foglight *for Azure SQL Database*

To generate reports for a selected database:

- 1 Go to **Dashboards > Reports**.
- 2 Click **Run a Report**.
The Run Report dialog box opens.
- 3 On the first screen, *Select Template*, Go to **All Templates**.
- 4 Select the requested report.
- 5 Click **Next**.
- 6 Select the requested time range.
- 7 Select the database or 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**.

The report generation process starts. Upon successful completion of this process, the Report Generated Confirmation dialog box opens.

- 13 Click **Download Now** to download the file.

Open the file using a relevant program for the selected format.

Reference

This section describes the collections that are used with monitored Azure SQL databases:

- [Overview of Collections](#)
- [SQL Performance Investigator Metrics](#)

Overview of Collections

Foglight *for Azure SQL Database* collects raw data for a set of collection types. This data is used by the Rules to trigger alarms and is used to populate dashboards. The collections data tables are not viewable from Foglight *for Azure SQL Database*, but the data is implemented in the various drilldowns of Foglight *for Azure SQL Database*.

i | **NOTE:** Several collections contain metrics whose name ends with Rate are not documented in this guide. They 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.

Usability

Purpose

Retrieves database usability based on response time and database availability.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	60
Online	300
Offline	60

Metric Descriptions

Metric	Description
Response Time	Average duration of the SQL statements that executed during the specified time range.
Up Since	The Azure SQL DB startup time.
Uptime	The amount of time Azure SQL DB was running since it was created.

Query Store

Purpose

Collects query store information.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	60
Online	300
Offline	900

Metric Descriptions

Metric	Description
Desired State	The desired operation mode of Query Store.
Actual State	The actual operation mode of Query Store. Difference between actual mode and desired mode indicates a problem.
ReadOnly Reason	When the desired_state_desc is READ_WRITE and the actual_state_desc is READ_ONLY, readonly_reason returns a bit map to indicate why the Query Store is in readonly mode. 1 - database is in read-only mode 2 - database is in single-user mode 4 - database is in emergency mode 8 - database is secondary replica (applies to Always On and Azure SQL Database geo-replication). This value can be effectively observed only on readable secondary replicas For other modes, refer to SQL online docs .
flush_interval_minutes	The period for regular flushing of Query Store data to disk in minutes. Default value is 15 minutes. Change by using the <code>ALTER DATABASE <database> SET QUERY_STORE (DATA_FLUSH_INTERVAL_SECONDS = <interval>) statement</code> .
interval_length_minutes	The statistics aggregation interval in minutes. Arbitrary values are not allowed. Use one of the following: 1, 5, 10, 15, 30, 60, and 1440 minutes. The default value is 60 minutes.
Storage Size	Size of Query Store on disk.
Max Storage Size	Maximum disk size for the Query Store.
current_storage_size_percent	Size of Query Store on disk in percentage.

Metric	Description
stale_query_threshold_days	<p>Number of days that the information for a query is kept in the Query Store. Default value is 30. Set to 0 to disable the retention policy.</p> <p>For SQL Database Basic edition, default is 7 days.</p> <p>Change by using the <code>ALTER DATABASE <database> SET QUERY_STORE (CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = <value>))</code> statement.</p>
Max Plans per Query	Limits the maximum number of stored plans. Default value is 200. If the maximum value is reached, Query Store stops capturing new plans for that query. 0 indicates there is no limitation with regards to the number of captured plans.
Query Capture Mode	<p>Actual capture mode of Query Store. can be:</p> <ul style="list-style-type: none"> • ALL - all queries are captured • AUTO - capture relevant queries based on execution count and resource consumption • NONE - stop capturing new queries. Query Store will continue to collect compile and runtime statistics for queries that were captured already
Cleanup Mode	<p>Actual size-based cleanup mode of Query Store. can be:</p> <ul style="list-style-type: none"> • OFF - no cleanup. • AUTO - size based cleanup will be automatically activated when size on disk reaches 90 percent of <code>max_storage_size_mb</code> • Size based cleanup removes the least expensive and oldest queries first. It stops when approximately 80 percent of <code>max_storage_size_mb</code> is reached
Capture Wait Stats	Indicates whether Query Store performs capture of wait statistics.
Storage Utilization (%)	Percentage of current storage size out of the <i>Max Storage Size</i> .

Query Store Top SQLs

Purpose

Collects query store top SQLs.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	300
Online	300
Offline	300

Metric Descriptions

Metric	Description
Query ID	Query ID
Interval ID	Runtime Stats Interval ID
Object Name	The database object that the query is part of. Ad-hoc query will be NULL.
Average Duration	Average duration for the query plan within @LastMinutes.
Average CPU Time	Average CPU time for the query plan within @LastMinutes
Average Logical IO Reads	Average number of logical I/O reads for the query plan within @LastMinutes
Average Logical IO Writes	Average number of logical I/O writes for the query plan within @LastMinutes
Average Physical IO Reads	Average number of physical I/O reads for the query plan within @LastMinutes
Average Memory Grant	Average memory grant for the query plan within @LastMinutes.
Average Tempdb Space Used	Average number of page reads for the query plan within @LastMinutes
Max DOP	Maximum DOP (degree of parallelism) for the query plan within @LastMinutes
Average Log IO	Average number of bytes in the database log used by the query plan, within @LastMinutes
Total Executions	Total count of executions for the query plan within @LastMinutes.
Plan ID	Plan ID.
Last Plan ID	Previous plan ID.
Total Duration	Total duration for the query plan within @LastMinutes
Total CPU	Total CPU time for the query plan within @LastMinutes
CPU Percentage	The CPU percentage of this query out of total CPU.
LOG IO Percentage	The LOG IO percentage of this query out of total LOG IO.
DATA IO Percentage	The DATA IO percentage of this query out of total DATA IO.

Version Info

Purpose

General Data about the database.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	3600
Online	3600
Offline	86400

Metric Descriptions

Metric	Description
Service Tier	The service tier for the database or data warehouse.
Service Tier Type	Purchase model. Can be DTU-based or vCore-based.
Pricing Tier	The pricing tier of the database.
Elastic Pool Name	The name of the elastic pool that the database belongs to.
Product Version	Version of the Azure SQL Database.
Product Level	Level of the version of the Azure SQL Database.
Collation	Name of the default collation for the server.
Version	Azure SQL DB version.

Resource Stat

Purpose

Collection of average CPU, I/O, and memory consumption for an Azure SQL Database.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	30
Online	60
Offline	300

Metric Descriptions

Metric	Description
Sample Time	UTC time indicates the end of the reporting interval.
Avg CPU (%)	Average compute utilization in percentage of the limit of the service tier.
Avg Data IO (%)	Average data I/O utilization in percentage of the limit of the service tier.
Avg Log Write (%)	Average write I/O throughput utilization as percentage of the limit of the service tier.
Avg Memory Usage (%)	Average memory utilization in percentage of the limit of the service tier. This includes memory used for storage of In-Memory OLTP objects.
Avg DTU (%)	Average DTU consumption in terms of a percentage of the maximum allowed DTU limit in the performance level for the user database.
Max DTU (%)	Maximum DTU consumption in terms of a percentage of the maximum allowed DTU limit in the performance level for the user database.
Max Workers (%)	Maximum concurrent workers (requests) in percentage of the limit of the database's service tier.

Metric	Description
Max Session (%)	Maximum concurrent sessions in percentage of the limit of the database's service tier.
XTP Storage Utilization (%)	Storage utilization for In-Memory OLTP in percentage of the limit of the service tier. This includes memory used for storage of the following In-Memory OLTP objects: memory-optimized tables, indexes, and table variables. It also includes memory used for processing ALTER TABLE operations. 0 if In-Memory OLTP is not used.
Samples	Number of samples.
DTU Limit	Current max database DTU setting for this database during this interval. NULL if database using the vCore-based model.
CPU Limit	Number of vCores for this database during this interval. NULL if database using the DTU-based model.

Wait Statistics

Purpose

Collects Database wait events grouped into resource categories.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	30
Online	60
Offline	300

Metric Descriptions

Metric	Description
Wait Type	Name of the wait type.
Wait Time	Total wait time for this wait type in milliseconds.

Database Connections Summary

Purpose

Collects and gather aggregated information about recent state of sessions connected to the Azure SQL DB.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	30
Online	60
Offline	300

Metric Descriptions

Metric	Description
Total Connections	This shows the total number of Azure SQL DB sessions. It includes user and system sessions.
Blocked Sessions	The total number of blocked sessions.
User Connections	This shows the number of Azure SQL DB user (non-system) sessions.
System Connections	The number of Azure SQL DB system sessions.
User Active Connections	The number of non-system sessions that are actively processing in Azure SQL DB or that are waiting on locks.
client_hosts	Number of total client machines.
All Active Sessions	The number of all Azure SQL DB active connections.
Lead Blockers	The total number of lead blocker sessions.
User Inactive Sessions	The number of non-system sessions that are not actively processing in Azure SQL DB.
Foreground Sessions	The number of foreground sessions.
Background Sessions	The number of background sessions.

Database Replication

Purpose

Collects Database replication status.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	60
Online	300
Offline	300

Metric Descriptions

Metric	Description
Partner Server	Name of the SQL Database server containing the linked database.
Partner Database	Name of the linked database on the linked SQL Database server.
Last Replication	The timestamp of the last transaction's acknowledgement by the secondary based on the primary database clock. This value is available on the primary database only.
Replication Lag (Sec)	Time difference in seconds between the last_replication value and the timestamp of that transaction's commit on the primary based on the primary database clock. This value is available on the primary database only.
Replication State	The state of geo-replication for this database, one of: <ul style="list-style-type: none">• SEEDING: The geo-replication target is being seeded but the two databases are not yet synchronized. Until seeding completes, you cannot connect to the secondary database. Removing secondary database from the primary will cancel the seeding operation.• CATCH_UP: The secondary database is in a transactionally consistent state and is being constantly synchronized with the primary database.• PENDING: This is not an active continuous-copy relationship. This state usually indicates that the bandwidth available for the interlink is insufficient for the level of transaction activity on the primary database. However, the continuous-copy relationship is still intact.
Geo-replication Role	Geo-replication role, one of: <ul style="list-style-type: none">• PRIMARY: The database_id refers to the primary database in the geo-replication partnership.• SECONDARY: The database_id refers to the primary database in the geo-replication partnership.
Last Commit	The time of last transaction committed to the database. If retrieved on the primary database, it indicates the last commit time on the primary database. If retrieved on the secondary database, it indicates the last commit time on the secondary database. If retrieved on the secondary database when the primary of the replication link is down, it indicates until what point the secondary has caught up.

Database Storage

Purpose

Collects Database storage information.

Collection Type

Azure SQL

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	60
Online	300
Offline	3600

Metric Descriptions

Metric	Description
Allocated	<p>The amount of formatted file space made available for storing database data.</p> <p>The amount of space allocated grows automatically, but never decreases after deletes. This behavior ensures that future inserts are faster since space does not need to be reformatted.</p>
Unused	<p>The difference between the amount of data space allocated and data space used. This quantity represents the maximum amount of free space that can be reclaimed by shrinking database data files.</p>
Used	<p>The amount of space used to store database data. Generally, space used increases on inserts. In some cases, the space used does not change on inserts or deletes depending on the amount and pattern of data involved in the operation and any fragmentation. For example, inserting one row does not necessarily increase the space used.</p>
Used Utilization	<p>Amount of used size as a percentage of maximum amount of space set to the database.</p>
Max Storage Size	<p>The maximum amount of space that can be used for storing database data.</p>

Database Recommendations

Purpose

Collects counter of all tuning recommendations of Azure SQL DB

Collection Sampling Settings

Frequency Mode	Collection Interval (Seconds)
Realtime Collection	60
Online	900
Offline	3600

Metric Descriptions

Metric	Description
Name	Unique name of recommendation.
Query ID	The query_id of the regressed query.
Type	The name of the automatic tuning option that produced the recommendation. For example, <code>FORCE_LAST_GOOD_PLAN</code>
Reason	Reason why this recommendation was provided.
Current State Reason	Constant that describes why the recommendation is in the current state.
Impact	Estimated value/impact for this recommendation on the 0-100 scale (the larger the better).
Script	Transact-SQL script that should be executed to force the recommended plan.
Reason Description	Describes why the recommendation is in the current state.
Regressed Plan ID	The plan_id of the regressed plan.
Recommended Plan ID	The plan_id of the plan that should be forced.
Regressed Plan Error Count	Number of errors of the query with regressed plan before the regression is detected.
Recommended Plan Error Count	Number of errors of the query with the plan that should be forced before the regression is detected.
Regressed Plan Execution Count	Number of executions of the query with regressed plan before the regression is detected.
Regressed Plan CPU Time Average	Average CPU time consumed by the regressed query before the regression is detected.
Recommended Plan Execution Count	Number of executions of the query with the plan that should be forced before the regression is detected.
Recommended Plan CPU Time Average	Average CPU time consumed by the query executed with the plan that should be forced (calculated before the regression is detected).
Valid Since	The first time this recommendation was generated.
Estimated Gain	Total CPU Time gain if we will apply the recommendation.
Error Prone	Yes, if <code>regressedPlanErrorCount > recommendedPlanErrorCount</code> .

SQL Performance Investigator Metrics

Active Connections

Number of all active connections, which were active during the current interval.

Active Time

Sum of all the active waits and CPU usage, equal to the session total activity within the current interval.

Active Time per Execution

Average Active Time per execution, during the current interval.

Active Time Rate

Rate of Active Time waits, during the current interval. Calculated in seconds/s.

Always On

Miscellaneous instance waits consisting of infrequent or otherwise special purpose wait states that should, in most cases, cases be very close to 0.

Average CPU Percent

Average percent of CPU time spent waiting for resources, like CPU and IO, out of the total active time.

Average DTU Percent

Average DTU consumption in terms of a percentage of the maximum allowed DTU limit in the performance level for the user database.

Average Memory Usage Percent

Average memory utilization in percentage of the limit of the service tier.

Average SQL Response Time

Average duration of the SQL statements, executed during the current interval. Calculated as `elapsed_time/num_executions`.

Average XTP Storage Percent

Storage utilization for In-Memory OLTP in percentage of the limit of the service tier.

Average Data IO Percent

Average data I/O utilization in percentage of the limit of the service tier.

Average Log Write Percent

Average write I/O throughput utilization as percentage of the limit of the service tier.

Backup Recovery

Time spent by the various sessions waiting for backup/recovery tasks to complete.

Blocked Sessions

Number of blocked sessions, during the current interval.

Connections

Number of Azure SQL DB sessions. It includes user and system sessions.

CPU Limit

Number of vCores for this database during this interval. NULL if database using the DTU-based model.

CLR Wait

Time spent by the various sessions waiting for CLR code execution to complete.

CLR Wait per Execution

Average CLR Wait per execution.

CLR Wait Rate

Rate CLR Wait. Calculated in seconds/s.

CPU Time

Time spent using CPU. Calculated in seconds.

CPU Time per Execution

Average CPU Time per Execution.

CPU Time Rate

Rate of time spent using CPU. Calculated in seconds/s.

CPU Wait

Time spent by the various sessions waiting in the system's run queue for CPU cycles.

CPU Wait per Execution

Average CPU Wait per Execution.

CPU Wait Rate

Rate of CPU wait. Calculated in seconds/s.

Cursor Synchronization

Time spent by the various sessions waiting for cursor synchronization operations to complete.

Database Replication

Time spent by the various sessions waiting for replication synchronization events to complete.

Deferred Task Worker

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.

Distributed Transaction

Time spent by the various sessions waiting for various distributed transaction events to complete.

DTU Limit

Maximum database DTU setting for this database during this interval. NULL if database using the vCore-based model.

Elapsed Time

Amount of time it took for the execution to end. This includes both active and idle time.

Executions

Number of times statements were executed which were active during the current interval.

External Procedure

Time spent by the various sessions waiting for external procedures to end.

Full Text Search

Time spent by the various sessions waiting for full text synchronization operations.

Governor Wait

The time spent by the various processes waiting for log rate governor operations to complete.

Governor Wait per Execution

Average Governor Wait per Execution.

Governor Wait Rate

Rate of Governor Wait. Calculated in seconds/s.

Granted Memory

Maximum amount of memory (KB) allocated to the execution of an activity.

Hosted Component

Time spent by the various sessions waiting for hosted components, such as, but not exclusively CLR.

Inactive User Connections

The number of non-system sessions that are not actively processing in Azure SQL Database.

I/O Bulk Load

Time spent by the various sessions waiting for a bulk load I/O to finish.

I/O Completion

Time spent by the various sessions waiting for I/O operations to complete.

I/O Data Page

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 waiting for physical input/output operations to complete.

I/O Wait per Execution

Average I/O Wait per Execution.

I/O Wait Rate

Rate of I/O Wait. Calculated in seconds/s.

Idle Time

Wait time for Idle.

Internal Cache Latch

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

Time spent by the various sessions waiting to latch a buffer that is not a physical I/O request.

Latch Savepoint

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.

Latch Wait per Execution

Average Latch Wait per Execution.

Latch Wait Rate

Rate of Latch Wait, Calculated in seconds/s.

Lead Blockers

The total number of lead blocker sessions.

Lock Bulk Update

Time spent by the various sessions waiting to acquire bulk update locks.

Lock Exclusive

Time spent by the various sessions waiting to acquire exclusive locks.

Lock Intent

Time spent by the various sessions waiting to acquire intent locks.

Lock Schema

Time spent by the various sessions waiting to acquire schema locks.

Lock Shared

Time spent by the various sessions waiting to acquire shared locks.

Lock Update

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.

Lock Wait Rate

Rate of Lock Wait. Calculated in seconds/s.

Lock Wait per Execution

Average Lock Wait per Execution.

Log Buffer

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

Time spent by the various sessions waiting for miscellaneous log waits.

Log Synchronization

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 LOG operations to complete.

Log Wait per Execution

Average Log Wait per Execution.

Log Wait Rate

Rate of Log Wait. Calculated in seconds/s.

Log Write

Time spent by the various sessions waiting for outstanding I/Os to finish, or waiting for log flushes to complete.

Logical Reads

Total number of logical reads operations performed by all requests running on the database.

Logical Reads per Execution

Average number of logical reads operations per execution performed by all requests running in the database.

Max Degree of parallelism

Maximum number of SQL Server threads assigned to the activity.

Max DTU Percent

Maximum DTU consumption in terms of a percentage of the maximum allowed DTU limit in the performance level for the user database.

Max Session Percent

Maximum concurrent sessions in percentage of the limit of the database's service tier.

Max Workers Percent

Maximum concurrent workers (requests) in percentage of the limit of the database's service tier.

Memory Wait

Time spent by the various sessions waiting for memory resources to be made available.

Memory Wait per Execution

Average Memory Wait per Execution.

Memory Wait Rate

Rate of Memory Wait. Calculated in seconds/s.

Network HTTP

Time spent by the various sessions waiting for outstanding HTTP connections to complete and exit.

Network I/O

Time spent by the various sessions waiting for network packets.

Network IPC

Time spent by the various sessions waiting for sub-tasks to generate data; long waits are indicative of unexpected blockages.

Network Mirror

Time spent by the various sessions waiting for database mirroring events to complete.

Network Wait

Time spent by the various sessions waiting for network I/O operations to complete.

Network Wait per Execution

Average Network Wait per Execution.

Network Wait Rate

Rate of Network Wait. Calculated in seconds/s.

OLEDB Provider Full Text

Time spent by the various sessions waiting for a remote OLEDB call to complete or DTS synchronization.

Other Miscellaneous

Time spent by the various sessions waiting for miscellaneous database operations.

Other Wait

Miscellaneous database waits consisting of infrequent or otherwise special purpose wait states that should, in most cases, be very close to 0.

Other Wait per Execution

Average Other Wait per Execution.

Other Wait Rate

Rate of Other Wait. Calculated in seconds/s.

Parallelism 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.

Percent of Total

Active time as a percent of the total active time of the instance in the same time frame.

Plan Recompilations

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.

Physical Reads

Total number of physical reads operations performed by all requests running in the database.

Physical reads per Execution

Average number of physical reads operations per second performed by all requests running in the database.

Preemptive Wait

Wait time for Preemptive mode.

Preemptive wait per Execution

Average Preemptive Wait per Execution.

Preemptive Wait Rate

Rate of Preemptive Wait. Calculated in seconds/s.

Remote Provider Wait

The time spent by the various processes waiting for a remote OLEDB call to complete or DTC synchronization.

Remote Provider wait per Execution

Average Remote Provider Wait per Execution.

Remote Provider Wait Rate

Rate of Remote Provider Wait. Calculated in seconds/s.

Row count

Number of rows that have been returned to the client.

Service Broker

Time spent by the various sessions waiting for Service Broker event handlers and endpoints.

Synchronous Task

Time spent by the various sessions waiting for tasks started synchronously (most SQL Server processes are started asynchronously).

Wait Time Percent

Percent of time spent waiting for resources, like CPU and IO, out of the total active time.

Writes

Total number of logical writes operations performed by all requests running on the database.

Writes per Execution

Average number of logical writes operations per execution performed by all requests running in the database.

XTP Log write Wait

Time spent waiting for in-memory log and checkpoint 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 Procedure Wait

Time spent waiting for events related to natively compiled stored procedures.

XTP Transaction Wait

Time spent waiting for in-memory transaction events.

XTP Wait

Time spent waiting for in-memory OLTP (XTP) related events.

XTP Wait per Execution

Average XTP Wait per Execution.

XTP Wait Rate

Rate of XTP Wait, Calculated in seconds/s.

Glossary

This section includes a glossary of terms used in this product.

Alarm

The mechanism by which Foglight *for Azure SQL Database* alerts users to a condition that might be a problem in the Azure SQL Database.

Authentication

Authentication is the process of proving users who they claim to be. A user connects to a database using a user account. When a user attempts to connect to a database, a user account and authentication information are required. The user is authenticated using one of the following two authentication methods:

- SQL Authentication.
With this authentication method, the user submits a user account name and associated password to establish a connection. This password is stored in the master database for user accounts linked to a login or stored in the database containing the user accounts not linked to a login.
- Azure Active Directory Authentication
With this authentication method, the user submits a user account name and requests that the service use the credential information stored in Azure Active Directory (Azure AD).

Calibration

The process by which Foglight *for Azure SQL Database* 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 Azure SQL Database* display the data flows correctly.

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. When a cartridge is installed and enabled, its components become part of the Management Server. Adding cartridges allows users to monitor additional parts of their environment. For further details about managing cartridges, See the *Foglight Administration and Configuration Guide*.

Client Machines

The machines on which the client executable (connected to Azure SQL Database) is running.

CPU Usage

When SQL statements and other types of calls are made to Azure SQL Database, 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.

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.

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 that the flow represents moves to another threshold, the flow may change color. The combination of movement and color makes it easy to spot congested areas. A graph above the flow shows how the load has varied over time.

Deadlock

A deadlock occurs when there is a cyclic dependency between two or more threads, or processes, for some set of resources within Azure SQL Database that is, each task has a lock on a resource which the other tasks are trying to lock.

When a database of the Azure SQL 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 Transfer Time

Reading or writing data requires a disk to access the disk sector where the requested data resides. After this sector is accessed, the amount of time required for a disk to read or write data from or to 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 during which a disk is busy servicing I/O requests.

DiskPerf

A Windows command line utility that enables or disables the collection of I/O statistics.

Dispatcher

Dispatcher is a background process, which is responsible for routing requests from connected user processes to available shared server processes, and for returning the responses back to the appropriate user processes.

The dispatcher process, which is only utilized with shared server configuration, handles and directs multiple incoming network session requests to shared server processes. At least one dispatcher process has to be created for every communication protocol available on the server.

Drilldown

A Foglight *for Azure SQL Database* view that provides more detailed information about a particular element (for example, a monitored object or an alarm) than the current view or dashboard. Foglight *for Azure SQL Database* drilldowns often contains charts or tables showing Azure SQL Database or Windows statistics or objects.

DTU

The Database Transaction Unit (DTU) is based on a blended measure of CPU, memory, reads, and writes. The DTU-based performance levels represent preconfigured bundles of resources to drive different levels of application performance. If you do not want to worry about the underlying resources and prefer the simplicity of a preconfigured bundle while paying a fixed amount each month, you may find the DTU-based model more suitable for your needs.

Elastic Pool

SQL Database elastic pool is a shared resource model that enables higher resource utilization efficiency, with all the databases within an elastic pool sharing predefined resources within the same pool. The workload pattern is well defined and is highly cost-effective in multitenant scenarios. Elastic pool is best for new SaaS apps requiring database isolation among tenants, or for modernizing existing apps to SaaS.

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 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.

Foglight Agent Manager

The Foglight Agent Manager is a client application that manages Foglight agents installed on monitored hosts. It provides a centralized communications link between the Foglight Management Server and the agents. The Foglight Agent Manager also provides a number of support services such as the ability to deploy, upgrade, and configure agents.

For further details, see the *Foglight Agent Manager 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.

General purpose

Ideal for most business workloads, offering balanced and scalable compute and storage options.

I/O Wait

I/O wait events take place when the session is waiting for the completion of input/output operations. The length of these wait events (the amount of time spent) depends upon the number of concurrent processes and threads requesting CPU time.

This metric's value should be inspected in conjunction with the value of the Run Queue Length.

Kernel Memory

The physical memory allocated to Windows kernel.

Kernel Mode

See [Process](#).

Kill

A Transact-SQL statement that terminates an Azure SQL Database connection. Any outstanding transactions for the selected session are rolled back and all locks are released.

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.

Logical Reads

A logical read occurs every time the Database Engine requests a page from the buffer cache.

LRU

Least Recently Used. Often refers to pages in memory. The LRU list is the mechanism by which Azure SQL Database 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 when a page was last modified.

Metric

A unit of measurement that can be applied to a database. Metrics can help gauge the performance of a system.

A metric is an individual piece of information that Foglight *for Azure SQL Database* collects about the performance of a system. The information may be a numeric value (a number or percentage), a string of text, or some other piece of data.

Every time the Foglight *for Azure SQL Database* dashboard is refreshed, the cartridge retrieves the latest value of the metric, which can then be displayed in a drilldown or on the home page.

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, which is measured in number (per second) of packets sent and received, 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

OLTP

Online Transaction Processing. OLTP allows real-time processing of SQL transactions, in order to support Customer Relationship Management (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 may include more than one company, OLTP databases use client/server processing and allow transactions to run on different platforms in a network.

Other Wait

Other wait events refer to time spent waiting for miscellaneous operations to complete. None of these operations fits into the separately identified wait event.

Paging

Disk I/O activity performed 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 Azure SQL Database* screens.

Physical Reads

A disk read I/O where the requester must wait for the disk read operation to complete. In Azure SQL Database, the requesting session waits while the page is read from disk. This is normally the most common type of read operation in Azure SQL Database.

Process

An instance of an application executing in Windows.

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.

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. It either prevents the deletion of primary keys if foreign keys exist, or deletes these foreign rows (DELETE CASCADE).

Relational Data Engine

A major functional part of Azure SQL Database. Responsible for the parsing and optimization of SQL requests, controls query plan execution, and processes row sets from the storage engine.

Schema Locks

A lock mode that is used when compiling a query, to prevent changes to the underlying tables structures while the query is compiled.

Severity

Represents how critical an alarm is. A severity determines how Foglight *for Azure SQL Database* 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 Azure SQL Database* takes when a metric value falls into the range defined by a threshold.

Single database

Single database offers provisioned compute and serverless compute tier choices.

SPID

Unique number the Server has assigned for identifying the session.

Spinner

A Spinner displays the current level of activity for a statistic that is not directional. As the load increases, so too does the speed of the spin. If the statistic represented by the flow crosses another threshold, the spinner may change color. The combination of movement and color makes it easy to spot congested areas.

Threshold

A range of values for a metric. If the metric falls within this range, Foglight *for Azure SQL Database* checks the threshold's severity to determine how to behave. For example, the component that represents this metric might change color.

Transaction

A group of one or more database modification statements that are combined into a logical unit of work that is either wholly committed or rolled back.

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 only be used by the table or index to which it is allocated.

vCore

A vCore-based purchase model is best if you are looking for flexibility, control and transparency of individual resource consumption. This model allows you to scale compute, memory, and storage based upon your workload needs and provides a straightforward way to translate on-premises workload requirements to the cloud.

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.