Foglight® for DB2 LUW 6.1.0.10

Monitoring DB2 Database Systems
User and Reference Guide

Contents

Introduction to Foglight for DB2 LUW

Welcome to the *Monitoring DB2 Database Systems User and Reference Guide*. This guide provides agent configuration instructions and information on investigating DB2 performance. It describes the dashboards included with the cartridge and contains instructions for configuring the agents the cartridge uses to collect monitoring data from the entire relational database management system. Also, this guide provides information about the rules and data tables that are available for your monitored system.

This guide is intended for any user who wants to know more about the agents properties and the dashboards included with the cartridge. It is also meant for those users that want to learn about the rules that are included with the cartridge, and the data the DB2 and DB2 Monitor agents collects.

Upgrading to the Current Version

Starting to work with a Foglight for DB2 instance requires upgrading to the current version of both the instance and the Agent Manager that runs the instance.

This topic contains instructions for using the upgrade wizard.

> **i** | **NOTE:** This topic is relevant only for existing Foglight users. If Foglight for DB2 LUW is being used for the first time and the Agent Manager is the updated version, as indicated in the release notes, skip this topic.

> **i** | **IMPORTANT:** Foglight for DB2 LUW does not support the upgrade method of placing the new version in the FGLHOME/upgrade/cartridge folder.

> **i** | **IMPORTANT:** When upgrading to the current version of Foglight for DB2 LUW in a Federation architecture, first upgrade the Federated database servers and then the Federation Server. If the Federation Server is upgraded first, the server displays incorrect information regarding the number of instances being monitored, as reported in the Status Summary. All the same, the information displays correctly on the Federated databases.

Overview

Any upgrade of Foglight for DB2 to the latest version, without upgrading the database cartridge components, is detected after the Databases dashboard is loaded. The upgrade is detected either automatically (by default: every 60 seconds) or manually, by clicking F5. Upon detecting the need for upgrade, the upgrade wizard starts automatically.
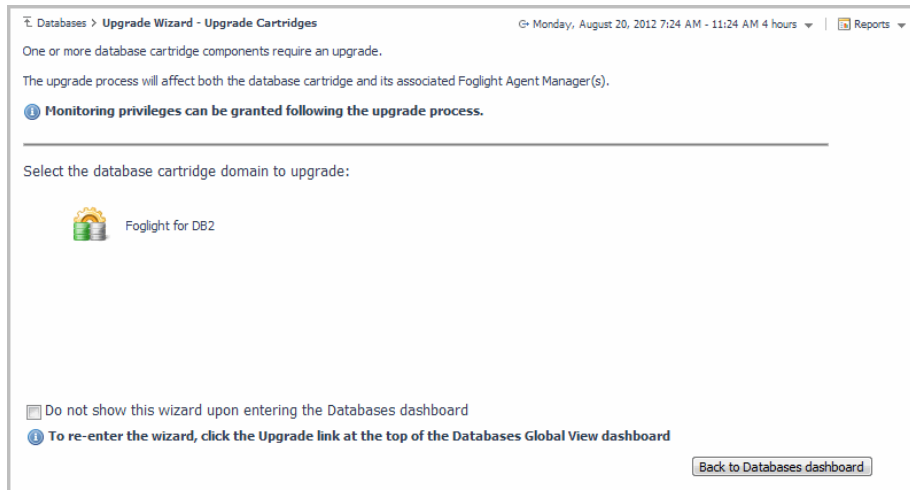
Selecting the check box at the bottom left of the upgrade wizard prevents the wizard from appearing upon entering the Databases dashboard. Nevertheless, if several database cartridge components still require upgrade, the need to upgrade them is still indicated on the dashboard by:

- The *Upgrade required* caption, which is displayed in red to the right of the requested instance

- The **Upgrade** button, which appears only if upgrade is required.

i | **NOTE:** Instances whose components must be upgraded appear in the Databases table. However, it is not possible to access those instances by clicking them. They must be upgraded first.

Running the Upgrade Wizard

To upgrade the requested instances:

1   Click **Upgrade**.

The screen that appears now is the same screen that appears by default upon entering the Databases dashboard when one or more database cartridge components require upgrading. The upgrade is carried out separately for each database cartridge type.

2   Click **Foglight for DB2**.

A Welcome page displays What's New in this release, including bug fixes. At the bottom of the *New Features* table, is a link to the Rule Modification Overview page, where you can review the list of rules that have been modified and are in the following states:

- Removed - The rule is obsolete and was deleted from the Management Server.

- New - The rule is new in this release. To review its definition, click View new rule.

- Updated - The rule was updated in this release. To review the updated rule, click View updated rule. If a rule targeted for update was modified by a user, the user's modified rule is copied and disabled before the updated rule is installed. To review the modified rule, click View user updated rule.

i | **TIP:** To avoid having to repeat rule modifications after an upgrade, do not edit rules with the DB2 prefix in the Rule Management dashboard. Use the Alarms Template dashboard instead. When you make your edits on the Alarms view, the edits are saved separately and applied over the predefined rules. For instructions, see Customizing Alarms for Foglight for DB2 LUW Rules on page 21.

3   Select the Agent Managers to upgrade..

> **NOTE:** The Other Agents column in the table indicates whether the specified Foglight Agent Manager runs agents of other cartridge types. If a value is displayed in this column, go to **Dashboards > Administration > Agents > Agent Status** to view which other agents run under the specified Foglight Agent Manager. If the other agents appear in the list below, upgrading Foglight Agent Manager to the latest version is safe. For agents of any other cartridge type, consult the documentation of the respective cartridge types.
>
> • DB_DB2_*
> • DB_SQL_Server_*
> • DB_Oracle_*
> • Sybase_MDA
> • UnixAgent
> • WindowsAgent

4   Click **Upgrade the selected Agent Managers**.

A progress bar appears.

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

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

privileges to ensure full functionality.

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

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

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

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

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

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

d   Click **Grant database privileges**.

Using ¹

Foglight *for DB2 LUW* monitors the DB2 database activity by connecting to and querying the DB2 database. The agents provided monitor the DB2 database system. The dashboards included with the cartridge provide a visual representation of the status of the major components of the DB2 agents. They allow determining any potential bottlenecks in database performance.

Viewing the Databases Dashboard

The Databases dashboard displays summary information about all databases that exist in the monitored environment.

To access this dashboard, from the navigation panel:

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

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

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

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

The Databases dashboard includes the sections listed:

*   Database cartridge type tiles — each tile represents a database type (SQL Server, Oracle, Sybase, DB2, or All) and displays the number of instances for each database type. The tile also displays a breakdown according to the instance health state severity (Normal, Warning, Critical, Fatal).

*   Status section — includes the following components:

    ▪   **Status summary** — a color-coded bar, which provides a visual representation of the summarized health condition of all instances listed in the Database Group table.

        The status summary bar provides a graphic representation of the monitored environment's current state, broken down to the number of instances and their current health state: Fatal, Critical, Warning, Normal or Unknown.

    ▪   **Database-specific health summary** — when the database group All is selected, this section displays all the currently monitored instances for each database type, divided by their health state (for example: four SQL Server instances, three of which have the health state Warning and one is indicated with the health state Fatal). When a user-defined database group is selected, this section displays data only about the agents included within the selected group.

        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 Selecting an Instance to Monitor on page 10.

    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 Selecting an Instance to Monitor on page 10.

1.

- The **Monitor** button — Use this button to add instances to 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 — Use this button to do one of the following:

    - Access the User Management settings, allowing you to restrict which instances specific users are allowed to view. This restriction makes it easier to for users to find information about only the instances they are interested in. For details, see Assigning Instances to Users on page 10.

    - 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.

- **Currently selected database group table** — a list of all monitored databases within the database group currently selected in the Databases section. For details, see Using the Currently Selected Database Group Table on page 13.

- The **Open the 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, by that means saving the need to navigate to these locations through the Overview drilldown.

i | **IMPORTANT:** If the display is filtered to show only DB2 instances the Databases table is renamed to DB2 Instances, and a new tab titled Reporting Services is shown. For details about this tab, see the following section.

Assigning Instances to Users

The User Level Access screen allows you to assign specific instances to users. As a result, when users view their Database dashboard, they only see the instances which have been assigned to them. This makes it easier to find the data which is relevant to them.

The User Management screen only displays users who are not Administrative users.

To assign instances to a specific user:

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

    The user management pane appears.

2   The table will display user names of all users, logon information, and instance information.

3   To assign one or more instances to a specific user, locate the user's name in the table, click **Configure** in the right-hand column, and then click **Manage** from the context menu.

4   In the Assign Instances view, select the instances which should be assigned to this user and click the > button to add the selected instances to the Assigned Instances column.

5   Click **Set** and **Close** to save your choices and exit.

Selecting an Instance to Monitor

Because the Databases dashboard displays by default all 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, DB2).

- **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 11.
- **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 11.
- **Assign Instances to specific users** — When accessing a Database dashboard, non-administrative users will view only instances which have been assigned to them. For details, see Assigning Instances to Users on page 10.

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.

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 a certain type of severity within a specific database type (for example: DB2 instances 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: **DB2** > **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.
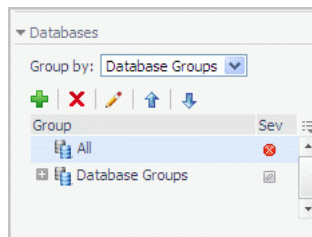
2   Click the ✚ button.

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:.

ℹ️ | **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 ❌ 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 DB2 LUW* Screens

The tables' **Customizer** button and the **In-context action** button are common to all Foglight *for DB2 LUW* screens, as described below.
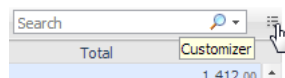
- The **Customizer** button — use this button, which is found on the upper right corner of each table, to create a custom filter for the table. The filter is created by clicking **Customizer** and specifying the criteria that the various values should meet in order to be displayed in the table (having 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*.

- **In-context actions** button — found on the upper right corner of all screens, and provides the following options:

  - **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).

  - **Refresh** — clicking this button retrieves the data acquired using the most recent sampling.

- **Useful links** — used for carrying out further investigation using the following external sources:

  - Navigate to Oracle Performance — a central knowledge base about all of Quest Software's database monitoring products.

  - Ask Foglight Community — Forum for Foglight products.

  - Contact support — central knowledge base about all of Quest Software's products.

  - About — Information regard the versions being used.

Using the Currently Selected Database Group Table

The currently selected database group table displays the columns listed below.

Name	Description
Instance	
Sev	The highest severity alarm of the DB2 database (whether a database instance, RAC or RAC node), which determines the database's health state.
Name	The name of the DB2 instance, RAC or RAC node. In addition, this row includes the **Go to Home Page** button.
Version	The version number of the DB2 database instance.
Up Since	The date and time that the DB2 database instance (or, in a RAC, the instance that was the earliest to restart) was last restarted.
Workload	The workload (average active sessions) for the DB2 database instance or RAC. 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 DB2 database instance/RAC. 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 instance or RAC.
CPU Load (%)	The overall operating system CPU usage by all processes (including CPU usage by the database).
Memory (%)	The percentage, within the total memory, of memory consumed by all operating system processes (including the database). This value includes both RAM resident memory and swapped memory.
Disk (% Busy)	The percentage of time the busiest device spent serving system-wide input/output requests. This metric serves as a measure for the system I/O load.

Name	Description
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.
	• Starting
	• Stopped
	• Stopping
	• Unknown
	When SQL PI is configured the SQL PI icon is displayed.
OS	The status of the OS data retrieval.
	NOTE: While the Foglight *for DB2 LUW* agent retrieves OS data, several OS metrics cannot be retrieved without the Infrastructure cartridge agent. IF OS monitoring was disabled during the creation of the Foglight *for DB2 LUW* agent through the Oracle Monitoring Installer wizard, the wizard cannot automatically create the Infrastructure cartridge agent; However, this agent can be created manually using the Dashboards > Hosts > Hosts dashboard. For details, see *Adding a Monitored Host* in the *Managing the Infrastructure Cartridge User and Reference Guide.*
	NOTE: If the Infrastructure agent was created, but OS database is collected only by the Foglight *for DB2 LUW* agent, click the OS status to OS data collection through the Infrastructure agent.

Adding and Configuring Agents

Enabling the Foglight Management Server to monitor DB2 instances requires the creation of the Foglight agents that monitor these instances. Ensure that these agents communicate properly with the Foglight Management Server.

Foglight *for DB2 LUW* 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. Foglight *for DB2 LUW* provides a wizard as a common entry point for adding and configuring instances for monitoring.

This topic ontains instruction for using the instance installation wizard, and the steps that should be taken for monitoring the newly added database instances.

> **i** | **IMPORTANT:** When running Foglight *for DB2 LUW* in a Federation architecture, agents cannot be created or administered 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).

Understanding the Foglight *for DB2 LUW* Architecture

The communication architecture of Foglight *for DB2 LUW*:

Figure 1. Communication architecture
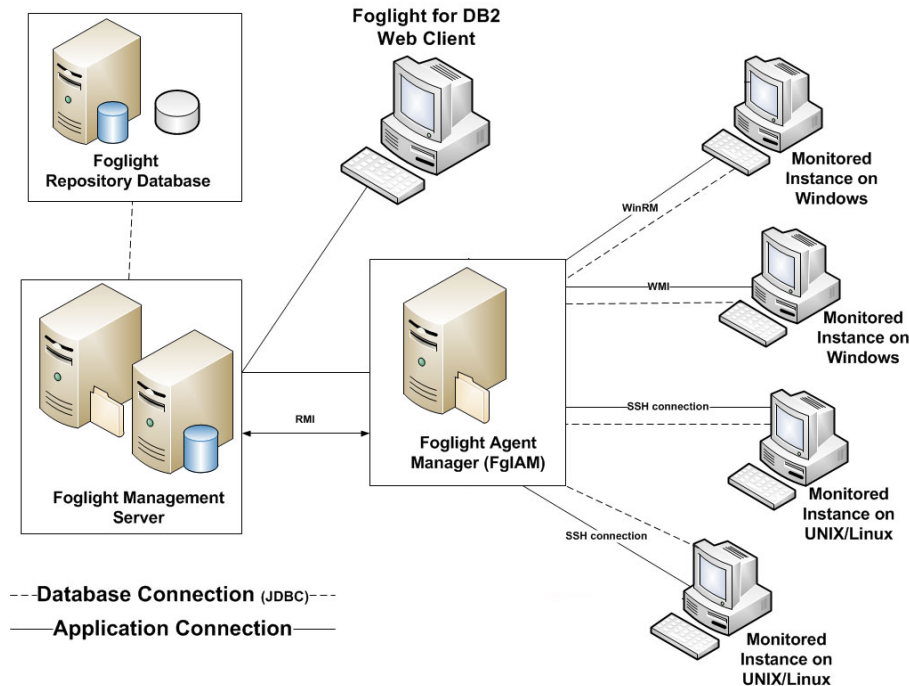
Configuring the On Demand Port on the Agent Manager Concentrator

If a firewall is installed between the Management Server and the Agent Manager, the bidirectional RMI connection is not possible because RMI requests require direct connection.

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

Figure 2. Bidirectional RMI connection.
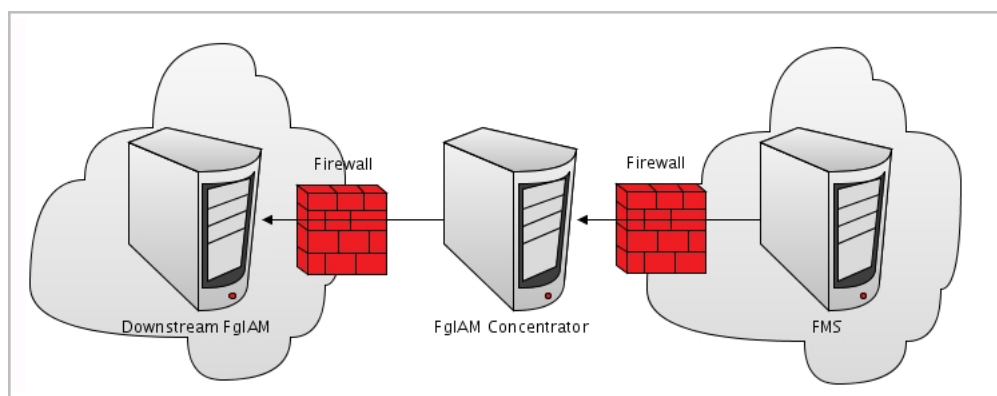
A concentrator agent, created specifically for Foglight, allows you to control the RMI server so that the server is initialized (starts collecting data) when the agent starts, and stops when data retrieval by the agent ends. The agent, created manually by the user, requires you to configure the following parameters:

- A list of the downstream hosts that are connected through the concentrator

- The RMI port number

Each time the Foglight Management Server sends an RMI request, the list of downstream hosts, which the concentrator agent submits, is being searched.

- Inclusion of the target RMI server host name within this list indicates that a FglAM concentrator exists, in which case a connection is made to the proxy RMI server.

- If the list does not contain the target host name, the connection is made directly to the target RMI server.

To create the concentrator agent:

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

2 Click **Create Agent**.

3 Select the agent type **DB_DB2_Concentrator**.

4 Enter a name of your choice.

5 Click **Create**.

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

To edit the concentrator agent connection details:

1 Select the newly created concentrator agent.

2 Click **Edit Properties**.

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

4 Enter the number of the proxy RMI server port, or accept the default number.

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

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

7 Click **Add Row**.

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

> **i** | **IMPORTANT:** When editing the properties of the Agent Manager concentrator (Dashboards > Administration > Agents > Agent Status), the name of the Agent Manager should be entered in the Downstream FglAMs section exactly as it appears in the topology, under Home > Agents > All agents > <Agent name> > RMI data > Fglam host property. The <Agent name> parameter is defined as any of the Foglight *for DB2 LUW* agents that reside on the selected Agent Manager.

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

10 Click **Save Changes**.

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

12 Click **Back To Agent Status**.

13 Select the newly created concentrator agent and click **Activate**.

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

> **i** | **IMPORTANT:** In an environment that includes an Agent Manager concentrator, when upgrading the environment where the concentrator does not contain any regular agents, the concentrator is not upgraded. Therefore, after the upgrade process takes place the Agent Manager concentrator must be deployed manually.

Configuring Foglight *for DB2 LUW* Monitoring Agents

A wizard guides you through the process of configuring Foglight *for DB2 LUW* agents. You are prompted to specify connection details for the DB2 host, instances, and databases. As you go, you can validate log in credentials, database and instance connectivity, and host connectivity. You can correct issues by granting database permissions, setting monitoring switches, and configuring parameters.

To monitor DB2 instances:

1   In the navigation panel, under **Homes**, click **Databases**.

    On a fresh installation, no instances are listed for DB2 in the Databases dashboard.

2   In the instances table, click the **Monitor** link and select **DB2**.

    The Monitor DB2 Instance wizard opens.

3   Follow the online prompts to configure an agent to monitor the DB2 host, instance, and databases. For help with options, click ⓘ .

Administering Foglight *for DB2 LUW*

You use the Databases Administration dashboard to set options for collecting, storing, and displaying data about monitored DB2 instances.

Opening the Databases Administration Dashboard

You can edit settings for one or more DB2 instances on the **Databases > Administration** dashboard.

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

To open the Databases Administration dashboard:

1   In the navigation panel, under **Homes**, click **Databases**.

2   Select the row check boxes beside one or more DB2 instances.

3   Click **Agent settings** and then click **Administration**.

    The Administration dashboard opens, containing settings for all the selected agents. Settings are broken down into categories, which are organized under a DB2 tree.

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

The Administration dashboard provides access to the following features:

- Reviewing the Administration Settings
- Customizing Alarms for Foglight for DB2 LUW Rules
- Defining Data Collection Options
- Database Connection Details
- Diagnostic Log
- On-demand Data Port

Reviewing the Administration Settings

The Databases Administration dashboard allows settings options for collecting, storing, and displaying data, which apply to all the currently selected agents. Click a category of settings on the left (for example: Connection Details) to open a view containing related settings on the right.

The metrics defined under the Databases Administration dashboard apply to all of the agents that were selected before opening the Databases Administration dashboard. As a result, the same unit of measure and aggregation value for display are enforced for all currently selected agents.

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

If the settings vary between the selected agents (for example: one agent uses the measurement unit kilobyte, while another uses megabyte), the fields that contain non-identical values are displayed as empty and marked with an Inconsistent Values ( ) icon.

Changes made to settings should be saved before selecting another category of settings.

To save changes made in an Administration dashboard view:

1   In the Database Administration dashboard, select a category from the menu.

2   Make changes to settings as necessary.

3   Click **Save changes** at the bottom of the view.

    If you attempt to exit the view without saving changes, a Warning dialog box prompts you to confirm your action.

    > **NOTE:** If you attempt to select instances of more than one type of database, such as an SQL Server database and DB2 instance, an error message is displayed.

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

Adding or Editing OS Credentials

The OS Connection Details page allows adding or editing the OS credentials for database instances and their hosts. Adding OS credentials for a database instance enables Foglight *for DB2 LUW* to monitor the instance's operating system.

> **CAUTION:** It is advisable not to modify existing credentials, because such changes affect all agents that share credentials.

Using Existing Host Credentials as OS Credentials

To use the host credentials to access the OS:

1   Select a database instance or host from the list.

2   Click **Set credentials**.

    The Edit Credentials dialog box opens.

3   Click **Use existing Host Connection Details** and click **Apply**.

Foglight *for DB2 LUW* attempts to connect to the OS. If the credentials are valid, the Save changes button is enabled. Otherwise, correct the credentials. See Creating or Editing OS Credentials.

4 When the credentials are valid, click **Save changes**.

Creating or Editing OS Credentials

To create or edit credentials used to access the OS:

1 Select a database instance or host from the list.

2 Click **Set credentials**.

The Edit Credentials dialog box opens.

3 Click **Edit Host Connection Details**.

The Edit Credentials dialog box expands to show the connection detail boxes.

4 If you want Foglight *for DB2 LUW* to monitor the instance's OS, ensure that the **Enable OS Monitoring** check box is selected.

5 If credentials exist in this lockbox or another lockbox, select the credentials.

    a Click **Select from stored credentials**.

       The Use Credentials dialog box opens.

    b Select a lockbox from the list.

    c Select a set of credentials from the list.

    d Click **Apply**.

The Edit Credential boxes display values for the selected credential. The password is masked.

6 Otherwise, create credentials by filling in the form.

    a Choose an authentication type from the list. The remainder of this example assumes a UNIX (log in cred.) authentication type.

    b Type a user name.

    c Type a password.

    d Type a port number.

    e If necessary, select the **Use Sudo** check box.

7 Click **Apply**.

Foglight *for DB2 LUW* attempts to connect to the OS. If the credentials are valid, the Save changes button is enabled. Otherwise, correct the credentials.

8 When the credentials are valid, click **Save changes**.

If an IC (Infrastructure Cartridge) agent currently exists, its credentials are updated. Otherwise, a new IC agent is created with the credentials.

Defining Data Collection Options

The Foglight for DB2 agent always collects and stores data, even when the browser window that displays the data is not active. Use the Collections page to specify:

• Which collections are sampled and stored.

• The data collection values when sampling is carried out in offline, online, and real-time frequency modes.

• The collection frequency.

When a user is focusing on a page, the sampling frequency for all the collections associated with this page automatically switches to Real-Time. The collection frequency setting determines the sampling frequencies of the other collections (collections that are not running in Real-Time mode).

The available collection frequencies can be selected from the *The selected agent's current collection frequency is (frequency)* field located in the middle of the panel.

To change the selected agent's current collection frequency:

1   Click agent's current frequency status.

    The frequency settings dialog displays.

2   Select the required frequency.

    The following settings are provided:

    ▪   **Low** — all collections are running in Offline mode, regardless of whether a Client is connected

    ▪   **Normal** — the collections running mode (Online/Offline/Real-time) adjusts dynamically to the Client's connection status (disconnected/connected/focusing on a page).

    ▪   **High** — all collections are running in Online mode, regardless of whether a Client is connected.

3   Select the required setting.

4   Click **Set**.

The Query timeout for on-demand collections setting defines the number of seconds that a query for on-demand collections runs before it times out.

On-demand collections are collections who retrieve data not by predefined time intervals but upon entering a page or clicking a button. The default setting of this parameter is 60 seconds. It can be modified by clicking the number that indicates the parameter's value in the field Query timeout for on-demand collections.

> **i** | **IMPORTANT:** In the Normal collection frequency, when the browser window that displays Foglight for DB2 is active, the collection frequency mode in the active page (for example: the Current Sessions panel in the Activity drilldown) switches to the fastest frequency possible, once every 20 seconds.

The Collections table contains the following columns:

Table 1. Collections table

Column	Description
Collection Enabled	Defines whether the selected collections are sampled and stored.
Collection Name	The name of the collection. This list is sorted in alphabetical order.
Offline Frequency (Sec)	Allows defining the collection interval, in seconds, in offline mode (for example: 300). Offline frequency is defined as the longest interval possible for sampling the monitored instance.
Online Frequency (Sec)	Allows defining the collection interval, in seconds, in online mode (for example: 60).
Real-time Frequency (Sec)	Allows defining the collection interval, in seconds, in real-time mode (for example: 20). Only one collection can be sampled for a real-time frequency at any given moment.
Query Timeout (Sec)	The amount of time, in seconds, that elapses before the query times out.

To modify the values of a specific collection:

1   Select the collection's row in the table.

2   Click **Edit**.

    The Edit the Collection dialog box appears.

3   Select whether to enable the collection and storage of the selected collection.

4   Set the collection interval, in seconds, in offline frequency mode (if available).

5   Set the collection interval, in seconds, in online frequency mode (if available).

6   Set the collection interval, in seconds, in real-time frequency mode.

7   Set the query timeout, in seconds.

8   Click **Set** to apply these settings or **Cancel** to reject them.

Customizing Alarms for Foglight *for DB2 LUW* Rules

> **i** | **NOTE:** Foglight 6.0 introduces Alarm Templates, which provide a simplified method for customizing alarm rules and applying them to agents. As part of this change, the Sensitivity Level feature has been deprecated.

Many Foglight *for DB2 LUW* 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
- Viewing Alarm Sensitivity Levels for Unassigned Agents
- Modifying Alarm Templates
- Configuring Email Notifications
- Cloning Agent Settings
- Reviewing Rule Definitions

Introducing the Alarms View

The Alarms view enables you to modify global settings and agent-specific settings for alarms.

To open the Alarms view:

1   Open the Administration dashboard as described in Opening the Databases Administration Dashboard on page 17.

2   Click **Alarms**.

The list of agents that you selected on the Databases dashboard is shown in the upper right corner of the view.

3   From the Alarms view, you can complete the following tasks:

- Viewing Alarm Sensitivity Levels for Unassigned Agents
- Modifying Alarm Templates
- Configuring Email Notifications
- Cloning Agent Settings

Viewing Alarm Sensitivity Levels for Unassigned Agents

In previous versions, Foglight *for DB2 LUW* offered four sensitivity levels that controlled which alarms were reported:

- **Essential** — Store and display only critical or fatal alarms.
- **Normal** — Store and display most alarms — essential and best practices; only critical and fatal statistical alarms.

- **Tuning** — Store and display all DB2 alarms sent to Foglight.

This feature has been deprecated in Foglight 6.0. These sensitivity levels can no longer be changed, but for agents that have not been assigned to an alarm template, you may view a report detailing the sensitivity levels previously configured.

To view the sensitivity level used by an agent:

i | **NOTE:** You may only view sensitivity levels for an agent that has not been assigned an alarm template.

1 In the Alarms view, click View sensitivity level as PDF in the upper right corner.

A **Rules Sensitivity Report** will be displayed.

Modifying Alarm Templates

Foglight 6.0 uses alarm templates to gather alarm rules into a domain-specific template that is easily modified and applied to targets. You can customize how the alarms generated by the default Foglight *for DB2 LUW* rules are triggered and displayed in by assigning alarm templates via the **Alarm Template Settings** tab. You can apply an existing template, create a new template using an existing template as reference, or create a template based on an agent. All changes to alarm templates apply to the selected agents.

Refer to **Viewing, Creating, and Managing Alarm Templates** in the *Foglight 6.1.0 User Guide* for more information.

i | **IMPORTANT:** Avoid editing Foglight *for DB2 LUW* rules in the **Administration > Rules & Notifications > Rule Management** dashboard. Default rules may be modified during regular software updates and your edits are lost. Always use the Alarm Templates dashboard.

Figure 3. Use the Alarms view to assign alarm templates.
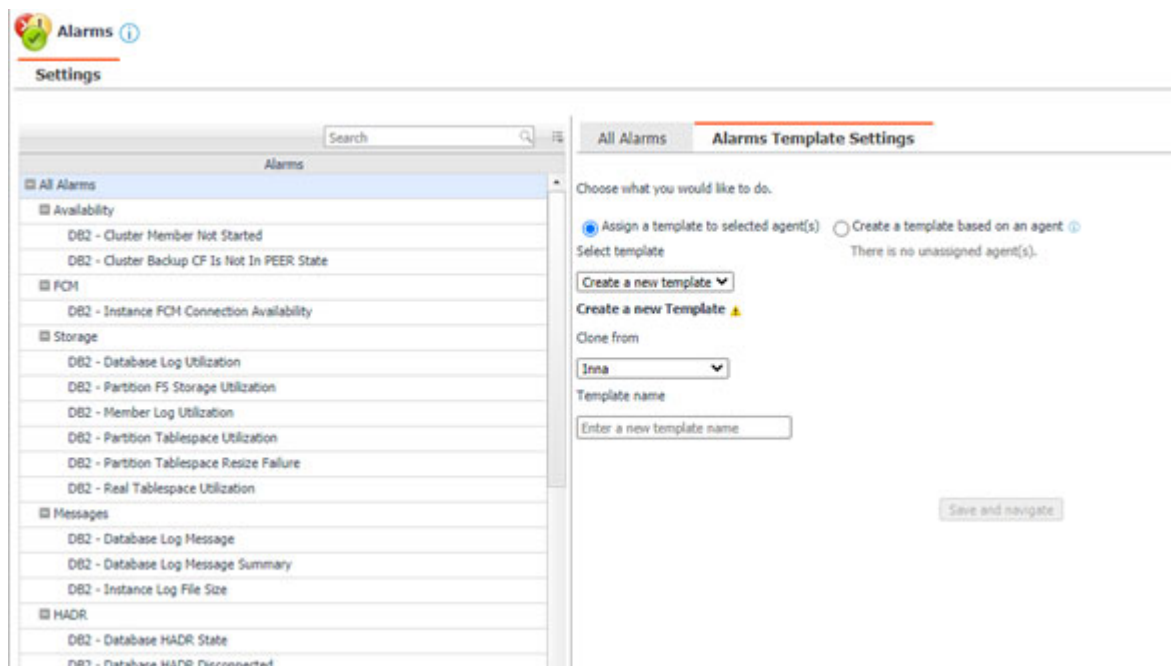
The Alarms list controls the contents displayed to the right and the tasks that are available.

- **All Alarms —** Displays all rules with configured alarms and indicates whether alarms are enabled. In this view, you can enable or disable alarms for all the rules at once. You can also set email notifications and define mail server settings.W hen viewing all alarms, the Alarm Template Settings tab is displayed, enabling configuration of alarm templates.

- **Category of rules —** Displays a set of related rules with configured alarms. In this view, you can set email notifications for the category of rules.

- **Rule name** — Displays the email notification status for the selected rule. If the rule has multiple severity levels, displays the notification configured for each severity level. In this view, you can enable or disable email notifications for the alarm and edit alarm messages.

- You can complete the following tasks:

- Assign an Alarm Template to selected agents

- Modifying alarm threshold values

Your changes are saved separately and applied over the default rules. This protects you from software upgrades that may change the underlying default rules.

Assign an Alarm Template to selected agents

You can override the alarm rules for the selected agents by assigning an alarm template. You can use the template to enable or disable alarms for all rules or an individual rule, or to change the threshold values of an alarm rule.

To see descriptions of the rules, follow the steps described in Reviewing Rule Definitions on page 28.

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. For an example, see the *DBSS - Worker Thread* rule. The threshold values corresponds to the lower bounds shown in this table. Many rules, such as Baseline rules, do not have severity levels and thresholds.

When editing thresholds, ensure that the new values make sense in context with the other threshold values. For most metrics, threshold values are set so that Warning < Critical < Fatal. However, in metrics where normal performance has a higher value, such as *DBSS - Buffer Cache Hit Rate*, the threshold values are reversed: Warning > Critical > Fatal.

To change alarm status and threshold values:

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 DB2 domain tab.

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 email notifications 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 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.

E.nabling 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.
- Changes in the monitored environment trigger an alarm.
- Alarm notification is enabled at the triggered severity level. See Defining email notifications, recipients, and messages.

To enable or disable email notifications:

1 In the Alarms view, click the **Settings** tab.

2   Decide on the scope for the change: all alarms, a category of rules, or a selected rule.

3   Complete the steps for the selected scope:

Table 2. 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 3. Configure email notification settings:

Scope	Procedure
All alarms	Click **All Alarms**. In the Alarms Settings tab, click the **Define Email Settings** button. Continue to Step 4.
Category of rules	Click a category. Click the **Define Email Settings** button. Continue to Step 4.
Selected rule	Click a rule. Click the **Define Email Settings** tab. • To change the severity level that warrants an email notification, click the link that displays the severities. Select the desired level of severity and click **Set**. • To configure email recipients and the message, select the tab for a severity level, and click **Edit**. Skip to Step 5.

4   If you selected **All Alarms** or a category, in the Email Notification Settings dialog box, do one of the following:

   ▪   To change the severity levels that warrant an email notification, from the **Messages will be enabled for severities** box, select the desired levels of severity.

   ▪   To configure the same email recipients and message for all severity levels, click **Configure mail recipients for all Severities** and then click **All severities**.

   ▪   To configure different email recipients and messages for each of the severity levels, click **Configure mail recipients for the following options** and then click a severity level.

5   In the Message Settings dialog box, configure the email recipients and message.

   ▪   **To** — Type the addresses of the people who need to act 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 Email Notifications on page 24.

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 26.

- **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 preceding the alarm information in the body of the email.

Figure 4. Message Settings.

6   Click **Set** to save the message configuration and close the dialog box.

7   If the Edit Notification Settings dialog box is open, click **Set**.

8   Click **Save changes**.

Defining variables to contain email recipients

You can create registry variables that contain one or more email addresses and (optionally) their scheduled notifications, and use these registry variables when defining email notifications. This procedure describes how to create a registry value. For schedules, see Defining scheduled email notifications.

To create a registry variable:

1   On the navigation panel, under Dashboards, click **Administration > Rules & Notifications > Manage Registry Variables**.

2   Click **Add**.

The New Registry Variable Wizard opens.

3   Select the registry variable type **String**, and click **Next**.

4   In the Name field, enter a name, for example: **EmailTeamName**

Optional — Add a description.

5   Click **Next**.

6   Select **Static Value**.

7   In the Enter desired value box, enter one or more email addresses (separated by commas).

>   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 25.

Defining scheduled email notifications

If someone wants to receive an email about an alarm regularly, such as once a day, you use a registry variable schedule to set up the notification.

To schedule the sending of email notifications for a registry variable:

1   If you are continuing from Defining variables to contain email recipients, the registry variable is already open for editing in the Edit Registry Variable dashboard.

>   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) to receive email notifications based on the schedule.

>   ℹ **TIP:** The addresses may the same as or different from the addresses 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.

> **i** | **IMPORTANT:** Avoid editing Foglight *for DB2 LUW* rules in the Rule Management dashboard. These rules may be modified during regular software updates and your edits are lost. Always use the alarm templates to modify rules.

You can create user-defined rules from the Rule Management dashboard. If you want to modify a rule, we recommend copying the rule and creating a user-defined rule. Manage user-defined rules from the Rule Management dashboard; these rules are not displayed in the Alarms view of the Databases Administration dashboard. For help creating rules, open the online help from the Rule Management dashboard.

To open the Rule Management dashboard:

1   On the navigation panel, under **Homes**, click **Administration**.

2   In the Rules & Notifications dashboard, click **Rules**.

3   Type **DB2** in the Search field to see the list of predefined rules for DB2 databases.

The Foglight *for DB2 LUW* 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.

7   Click **Cancel** to avoid changing the rule unintentionally.

Adding User-defined Collections

This page allows you to add user-defined collections to all the currently selected agents and to provide for queries not included in Foglight for DB2.

> **i** | **IMPORTANT:** Agents must be enabled for user-defined collections. If the requested agent is not enabled for such collections, go to the Connection Details page and click the agent to edit it.

After collections are added, this page displays all the user-defined collections for all the agents. For example, if a collection was added to 12 agents during its addition, the page displays 12 rows, showing the collection for each agent.

This page is also used for configuring the sampling frequency for each collection. The available sampling frequencies are as follows:

•   **Real —** a user is focusing on a page, the sampling frequency for all the collections associated with this page switches to Real-Time.

•   **Online** — when at least one user is connected, the sampling frequency for all the collections that are not currently running at Real-Time frequency switches to Online.

- **Offline** — when no user is connected to the application

To add user-defined collections:

1   Click **Add**.

The *Add a User-defined Collection* dialog box appears.

2   Click **Set details**.

The Set Collection Details dialog box appears.

3   Enter a name in the Collection name field.

4   Enter a brief description of the collection in the Collection description field.

5   Paste the query's SQL text in the Query Text field.

> **i** | **NOTE:** This field can hold up to 4096 characters.

6   Type a value, in seconds, in the Query Timeout field.

7   Click **Verify**.

The collection is verified by running the query on each of the currently selected agents. After the verification process is complete, the Verification Results pop-up appears, indicating whether the collection was verified successfully. In case the collection verification failed, the relevant error message is displayed. If the verification succeeded on at least one agent, the collection's details are displayed.

Table 4. Collection details

Section	Field	Description
Database	**ID**	The collections' ID
	Column Name	The name of the column
	Column Type	The field type.(String, Integer, etc.)
Data Storage	**Display Name**	The column's display name
	Type	The topology type for storage purposes. This type can be one of the following: • String • Integer • Double • Boolean • Date **NOTE:** When the Frequently modified check box is selected (the default), the field's change history is kept. This includes use of optional functionality such as use of the IntelliProfile mechanism and aggregation type selection. Clearing this check box is recommended only for fields whose values change infrequently, such as IP address of a specific host. Storing the change history of such fields is highly CPU-intensive and may degrade the FMS performance.
	Unit of Measurement	The metric's unit of measurement. The possible measurement unit values are: Percent, Count, Millisecond, Second, Minute, Hour, Day, Byte, Kilobyte, Megabyte and Gigabyte.

Table 4. Collection details

Section	Field	Description
	Aggregation	Allows defining the value that is displayed in this metric, out of several values that were returned in the relevant time range. The available aggregation values are: • **As is** — value of the last sample taken during the selected time range. • **Sum** — summarized value of all samples taken during the selected time range. • **Average** — average value of all samples taken during the selected time range.
	Is Key	Indicates whether the field is the query's key for retrieval. **NOTE:** Fields that are indicated as keys should correspond with the database result set unique values. While a field whose results are frequently changing and repeating can be selected as key, such a selection may result in unexpected behavior from the defined collection.
	Configured to use IntelliProfile	Select **True** in this column to enable the collection of baseline statistics for this metric.

8   The Edit Sampling Frequencies section allows you to change sampling frequencies. Click the required frequency.

The *Edit Frequencies for the Collection* dialog box appears.

9   Enter the required changes.

10 Click **Set**.

11 Click **OK**.

> **i** | **IMPORTANT:** After adding the collection by clicking OK, you will not be able to make changes except to modify the collection frequencies and enable or disable the collection.

On-demand Data Port

The On-demand Data Settings page allows you to define a port for each Foglight Agent Manager. Foglight for DB2 uses this port to retrieve data and for integration with StealthCollect. By default, the collection of on-demand data is carried out using the same port used for connecting to the Performance Analysis middleware. Use this page to collect on-demand data through a different port, or if another port was allocated to the Performance Analysis middleware.

To configure a port for the collection of on-demand data:

1   Select the check box near the names of the requested Foglight Agent Managers.

2   Click **Set port**.

The Set On-demand Data Port dialog box appears.

3   Type the number of the requested port.

4   Click **Set**.

5   Click **Validate connectivity**.

The Validate connectivity progress bar appears. At the end of the validation process, the connection status appears on the Status column.

ℹ IMPORTANT: Changes to the On-demand Data Port page take effect immediately and do not need to be saved.

Database Connection Details

You use the Database Management view on the Databases Administration dashboard to manage the list of monitored databases and validate their connection settings.

Setting credentials and instance properties

You can validate the login credentials and instance properties.

To edit instance credentials and properties:

1 Select the instance you want to set the credentials for.

2 Click **on the instance name.**

   The DB2 Instance details dialog appears.

3 Edit the details.

4 Click **Validate**.

To edit database credentials:

1 Select the required database.

2 Click on the Validate connectivity link in the connection status beside the database name.

3 Select whether to use the same credentials used for the instance (default), or to assign custom credentials for the database.

   ℹ TIP: An instance must have at least one monitored database. If an instance contains no databases you want to monitor, remove the instance.

Diagnostic Log

Use the Diagnostic Log view to configure the retrieval and display settings for the Log panel. In addition, this view is used to define default settings for invoking alarms based on the collected alerts.

Use this view to define the following settings:

- Minimal severity for invoking summary alarms - can be either turned off (the default setting) or set to one of the defined severity values, that is: Warning, Critical, Error, Event or Severe.

- Minimal severity for invoking alarms - can be either turned off or set to one of the defined severity values: Warning, Critical, Error, Event or Severe (Default).

- Number of rows to collect from logs - the maximum number of rows that would be retrieved from the alert logs.

- Messages to include in the Log panel display - using the Match List pane, which contains a default list of expressions within the diagnostic log that are to be retrieved and displayed in the Log panel.

- Message to Exclude from the Log panel display - using the Ignore List pane, which contains a default list of expressions within the diagnostic log that are to be excluded from the diagnostic log display.

ℹ IMPORTANT: Only messages that are explicitly defined in the Match List and not filtered in the ignored list will be displayed.

Both the Match List and the Ignore List panes can be customized by adding, editing or removing messages. Each filter can be enabled or disabled separately by clicking **Edit** and selecting or clearing the Enabled check box. Alternatively, to enable or disable all of the filters, click **Enable All** or **Disable All**.

To add a message to the Match List:

1 Click **Add**.

   The Add an Aleart to the Match List dialog displays.

2 Use this dialog box to configure the alert filter by means of the following fields:

   ▪ **Filter Name** - used for assigning a name to the filter.

   ▪ **Field Name** - used to select by which of the message parameters, message will be filtered. Possible fields include: Severity, Message text, Impact, Component, Function, Type or FullRec (The entire message record)

   ▪ **Expression** - identifies a regular expression that will be evaluated against the Field selected above and used together in order to select the message. For example: To filter message that their text related to files, you should select "Message" as the Filter Name and "*File*" as the Expression.

3 Click **Add** to save your settings. Each newly added alert filter is enabled by default.

   To disable the filter, click **Edit** and then clear the Enabled check box.

To edit a Filter:

1 Click the requested filter.

2 Click **Edit**. The Edit the Selected Alert dialog opens.

3 Use this dialog to configure the alert by using the same fields when you added it: Filter Name, Field Name, and Expression.

4 To disable the filter, clear the Enabled check box.

5 Click **Set** to save your settings.

6 At the bottom of the view, click **Save Changes**.

Exploring DB2 Database Performance Metrics

The Foglight for DB2 Foglight *for DB2 LUW* provides a set of homepages that display performance metrics collected from the monitored systems. These homepages allow you to monitor, analyze, investigate the DB2 performance, and identify potential bottlenecks. The various homepages are described in the following sections and include a reference with the heading and page number:

- Instance Homepage
- Database Homepage
- Member Overview Homepage
- Member (Real Time) Homepage

Instance Homepage

The DB2 Database Instance Homepage displays a summary view of the instance health. You drill down further to carry out root cause analysis of various performance issues.

To access the Instance Homepage, drill down to **Databases Dashboard > Instance Homepage** ☑ .

Figure 5. Instance Homepage button
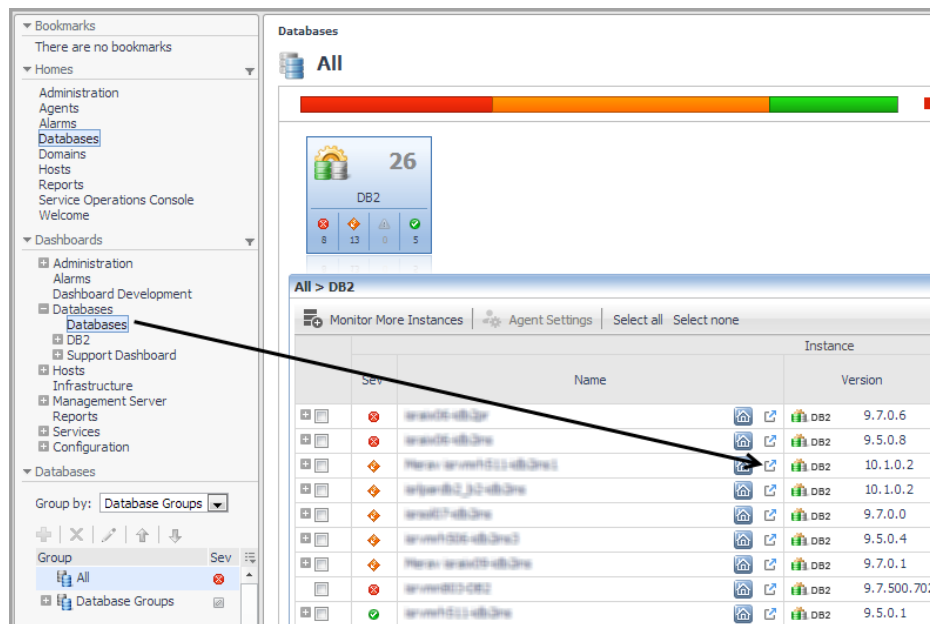
Figure 6. Instance Homepage
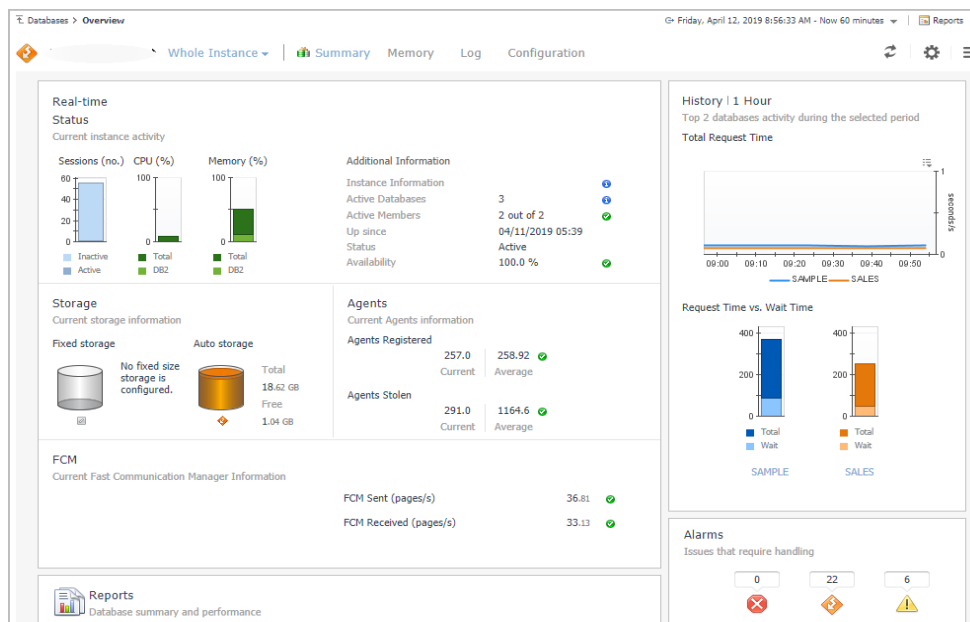
For more information, see

Database Homepage

The Database Homepage provides an overall view of the status of the selected entity for the selected time range, as well as drill-downs, which provide a visual representation of the status and performance of the monitored DB2 environment. These drill-downs allow you to determine any potential bottlenecks in database performance.

To access the Database Homepage, navigate through **Databases Dashboard** > **Instance Homepage** > **Select Database**, and then select a database name.

You can also access the Database Homepage for a database from the Databases Dashboard by clicking the icon at the far right of the database name cell and then selecting **Database Homepage** from the menu that appears.

> **i** | **NOTE:** The Infrastructure agent configuration indicator at the bottom right of the Homepage notifies if the Infrastructure agent is installed and configured for the selected instance. This agent enables OS monitoring of the hosts where the DB2 LUW instance resides.

For information about the various sections appearing on this Homepage, see Database home page on page 51.

Using the Database Homepage Toolbar

The Database Homepage toolbar allows you to drill down and view a variety of DB2 database metrics.

Figure 7. Database Homepage toolbar
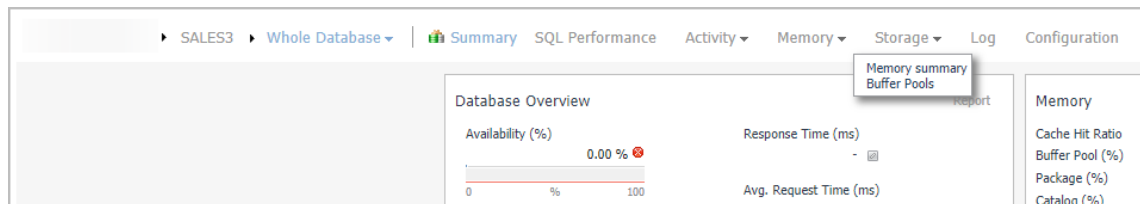
The following toolbar options are provided:

- **Summary** — goes back, from any of the drill-downs, to the Database homepage.
- **SQL Performance** — provides the ability to investigate the activity and resource consumption of a selected database or database partition. For more information on this dashboard, see SQL Performance on page 55.
- **Activity** — allows you to explore SQL metrics using the following panels:
    - Top Tables Panel
    - Locks Panel
    - Current Agents Panel
    - User-defined Collections Panel
- **Memory** — allows you to explore database memory using the following panels:
    - Memory Panel
    - Buffer Pools Panel
- **Storage** — allows you to explore DB2 storage using the following panels:
    - Tablespaces Panel
    - Tablespaces Bufferpool Activities Panel
- **Log** — allows you to explore the Diagnostic Log Panel.
- **Configuration** — allows you to explore the Initialization Parameters Panel.

Activity Panel

Top Tables Panel

Use the Top Tables panel to help you monitor the performance of database tables.

To access the Top Tables panel, navigate through **Databases Dashboard > Instance Homepage > Database Homepage**, and then choose **Activity > Top Tables**.

For more information about this panel, see Top Tables on page 63.

Locks Panel

The Locks panel displays information about current locks and historical locks held in the selected database.

To access the Locks panel, navigate through **Databases Dashboard > Instance Homepage > Database Homepage**, and then choose **Activity > Locks**.

You can also access the Locks panel for a database from the Databases Dashboard by clicking the icon at the far right of the database name cell and then selecting **Locks** from the menu that appears.

For more information, see Locks on page 65.

Current Agents Panel

This panel provides metrics on agent workload, activity, and background operations.

To access the Current Agents panel, navigate through **Databases Dashboard > Instance Homepage > Database Homepage**, and then choose **Activity > Current Agents**.

You can also access the Current Agents panel for a database from the Databases Dashboard by clicking the icon at the far right of the database name cell and then selecting **Current Agents** from the menu that appears.

For more information, see Current Agents on page 67.

User-defined Collections Panel

The User-defined Collections panel allows viewing the customized collections, which were created via the User-defined Collections page in the Administration dashboard, either during the last sample (Last Snapshot) or plotted over the specified time range (Selected Period).

To access the User-defined Collections panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Activity > User-defined Collections**.

For more information, see User-defined Collections on page 70.

Memory Panel

Memory Summary Panel

The Memory Summary panel provides memory summary metrics on the members and memory pools for the selected database.

To access the Memory Summary panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Memory > Memory summary**.

For more information, see Memory Summary Pane on page 73.

Buffer Pools Panel

The Buffer Pools panel provides performance metrics on the buffer pools for the selected database.

To access the Buffer Pools panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Memory > Buffer Pools**.

For more information, see Buffer Pools Pane on page 74.

Storage Panel

All database data is stored in a number of Tablespaces. There are different types of tablespaces, including catalog, regular, large, system-temporary, and user-temporary tablespaces. Tablespaces can be managed by the OS (System-Managed Space — SMS) or the database (Database-Managed Space — DMS). Each tablespace can have one or more containers. Each tablespace has a page size associated with it, which limits the row length and

column count of tables that can be placed in the tablespace, meaning that a larger page increases the capacity of the tablespace. The utilization of the page size has an effect on the usage of overall tablespace resources. For example, having a small table with a large page size results in a significant amount of wasted space and inefficient I/O and buffering usage.

Tablespaces Panel

The Tablespaces panel shows the top ten tablespaces with the highest percentage of the space usage, along with individual table properties for each individual tablespace. Use this panel to quickly investigate the existing tablespace usage and look for parameters that can indicate potential performance issues.

To access the Tablespaces panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Storage > Tablespaces**.

You can also access the Tablespaces panel for a database from the Databases Dashboard by clicking the icon at the far right of the database name cell and then selecting **Tablespaces** from the menu that appears.

For more information about this panel, see Tablespaces on page 77.

Viewing individual tablespace details

The Tablespace Details view shows individual tablespace properties and metrics. This dashboard also provides space utilization and free space fragmentation charts that can give you insight in how well the system can perform based on the existing storage resources. For example, a high amount of fragmented disk space can lead to increased performance overhead. You drill down to this view by clicking a tablespace entry on the Tablespaces dashboard or clicking on the tablespace name in the Tablespace Summary section.

Figure 8. Drill down details
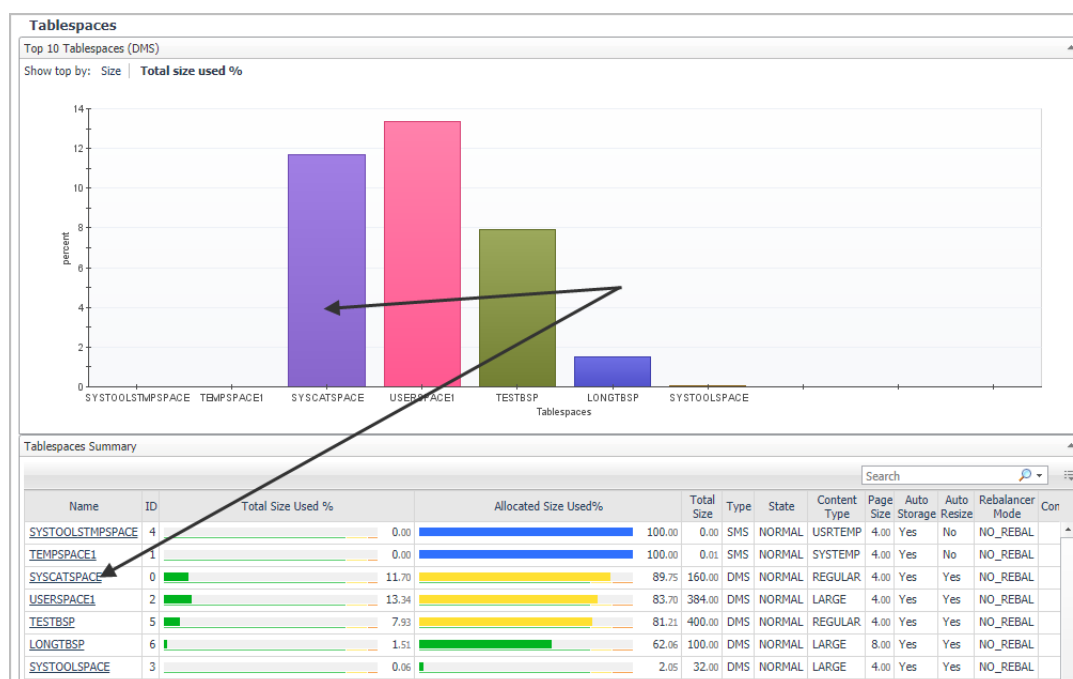
For more information about this view, see Tablespace Details on page 84.

Tablespaces Bufferpool Activities Panel

A buffer pool is memory that temporarily stores data and index pages from the database. If an application needs to read a data or index page and that page is already in the buffer pool, the application reads the page from the buffer pool, a process that is faster than retrieving the page from disk. Avoiding disk I/O is a critical issue when you are trying to improve database logical and physical read performance.

The Tablespaces Bufferpools Activities panel shows the top tablespaces with the highest percentage of the space usage, and for each tablespace, it displays its buffer pool usage, such as data reads and writes. Use this

dashboard to quickly investigate the buffer pool usage for the top tablespace consumers, and to look at the overall buffer pool performance.

To access the Tablespaces Bufferpool Activities panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Storage > Tablespaces Bufferpool Activities**.

For more information about this panel, see Tablespaces on page 77.

Diagnostic Log Panel

The Diagnostic Log panel provides a graphic, color-coded representation of the notification log messages, sorted by their severity level, as detailed below. Informational messages and warnings are displayed here, but do not invoke alarms.

To access the Diagnostic Log panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Log**.

For more information, see Log Drill-down on page 88.

Initialization Parameters Panel

The Initialization Parameters panel provides the list of initialization parameters with their current values and other pertinent information, sorted in alphabetical order.

To access the Initialization Parameters panel, navigate through **Databases > Instance Homepage > Database Homepage**, and then choose **Configuration**.

For more information, see Configuration Drill-down on page 89.

Member Overview Homepage

The Member Overview Homepage provides metrics on the state of the member over the course of the selected time range. Popups provide graphs that show you how the member is performing and whether or not there are any critical areas that need to be addressed.

Figure 9. Member Overview Homepage
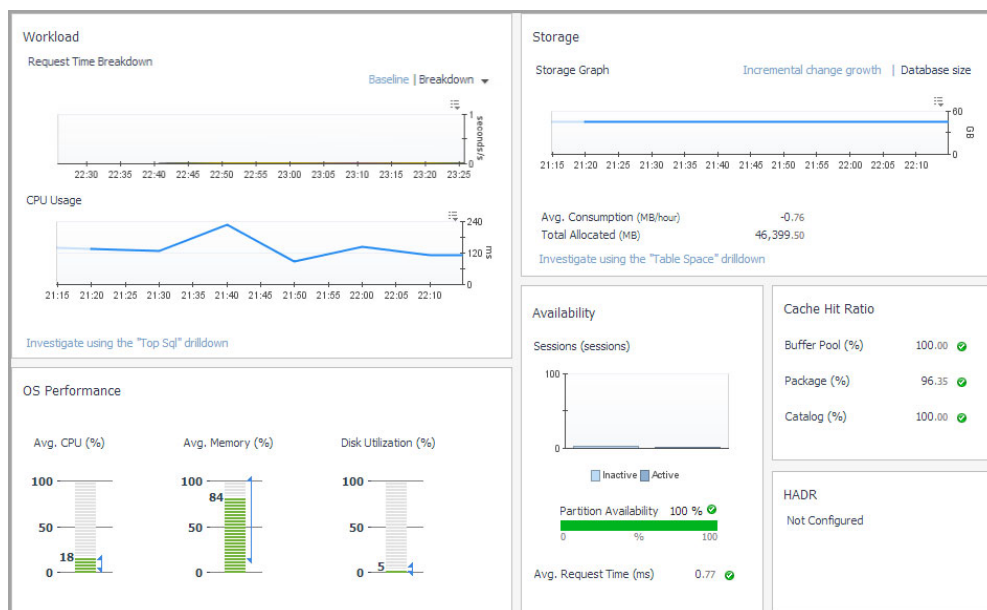
For more information, see Member Overview home page on page 90.

Member (Real Time) Homepage

The Member (Real Time) Homepage provides metrics on the present state of the selected member. It allows you to view how the member is performing and any performance or bottleneck issues affecting the member. Popups provide additional information and graphs.

Figure 10. Member (Real Time) Homepage
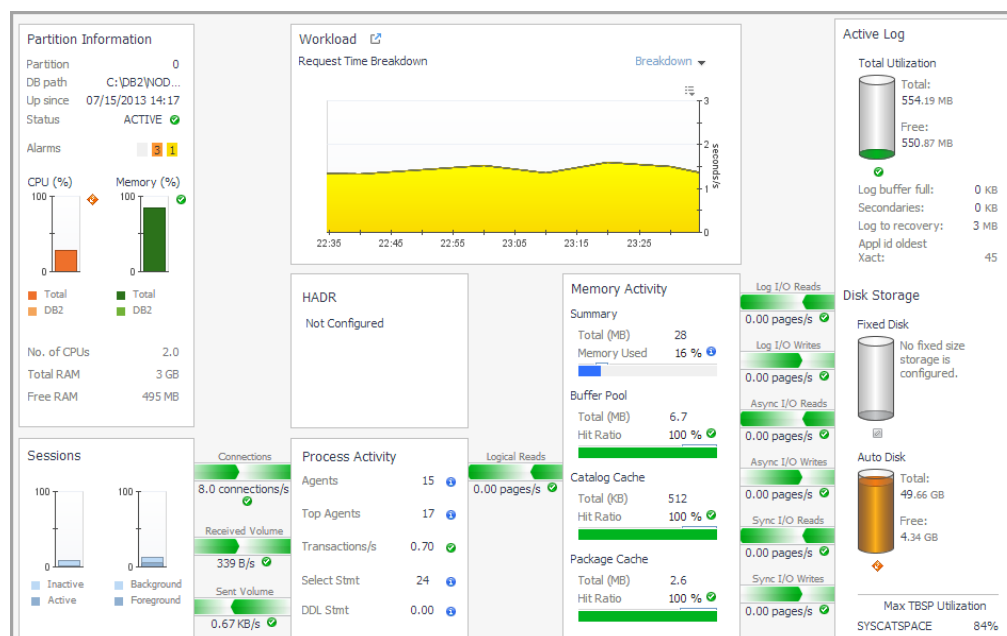
For more information, see Member (Real Time) home page on page 94.

Reporting on Your DB2 Environment

Foglight *for DB2 LUW* allows you to generate reports about various aspects of the performance of the selected database instance. This section provides instructions for generating reports and a brief description of each report.

Foglight *for DB2 LUW* comes with a set of pre-defined reports. Operators can run those reports, build custom reports, and view generated reports. Advanced Operators can also schedule and manage reports.

To access the Reports dashboard, from the Foglight navigation pane, select **Dashboards > Reports**. From the Reports dashboard, click the links provided as described below:

- Click **Build a Custom Report** to choose the building blocks for your custom report using the Create Report wizard.

- Click **Run a Report** to select and generate or schedule a report using the Run Report wizard.

- Click **Manage Reports** to access the Manage Reports dashboard, where you can download, delete and view details about generated reports.

ℹ | **NOTE:** You can also access the reports features using the Reports menu at the top right of any dashboard or on the General tab of the Action panel to the right of any dashboard. Select the **Create a new report** option to launch the Create Report wizard and define the components to be used to build a new report.

For more information on using the Foglight reporting features, see the *Foglight User Guide or* online help.

About the Foglight *for DB2 LUW* Reports

Foglight *for DB2 LUW* comes with a set of pre-defined report templates, which provide an analysis of the performance and health of your DB2 LUW environment over time.

Each report template uses report parameters to define the content of the report. Use a template's default report parameters to quickly generate or schedule a report, or optionally change the report parameters to customize the report to better meet your requirements.

The following Foglight *for DB2 LUW* reports are available:

> **i** | **NOTE:** If you are using the Templates by Module tab on the Run Report wizard, click the expansion box to the left of Databases > DB2 to view the Foglight *for DB2 LUW* reports.

- Comparison of Databases Report — reports on and compares the I/O activity between selected databases. For this report, you provide a time range and more than one database.

- Executive Workload Summary Report — provides a workload overview for the instance and for each database.

- Health Check Report — reports on the database availability, response time, and connection time. For this report, you provide a time range and a database.

- Storage Summary Report — provides an overview of storage utilization using relevant categories such as disk space usage, file system usage (as it relates to automatic storage tables spaces), log storage, and top-consuming tablespaces. For this report, you provide a time range and a database.

- Top SQL Report — provides detailed information regarding the SQL statements that experienced the longest time of CPU consumption or total wait events.

Reference

This topic contains reference information about the following topics:

- Dashboards and Views
- Collections and Metrics

Dashboards and Views

Foglight displays monitoring data in views that group, format, and display data.

Dashboards are top-level views that contain lower-level views. The dashboards supplied with Foglight, and user created dashboards, are accessible from the navigation panel.

Lower-level views in Foglight can be added to dashboards or can be accessed by drilling down from a dashboard. They receive and display data directly from the Foglight Management Server or from other views. Some views filter or select data that appears in other views in the same dashboard. Some are tree views with expandable nodes for selecting servers, applications, or data.

This section describes the following Foglight *for DB2 LUW* dashboards and home pages:

- Databases Dashboard
- Instance home page
- Database home page
- Member Overview home page
- Member (Real Time) home page

Databases Dashboard

The Databases dashboard contains the following elements:

- Database Alarms
- Database Environment Tiles
- Databases Instances View
- Database Cue Card

Database Alarms

The Database Alarms pane shows the numbers alarms, by the severity level (Fatal, Critical, Warning, Normal, and Unknown). Use it as a starting point to quickly identify the sources of problems within your database infrastructure.

Figure 1. Database Alarms pane
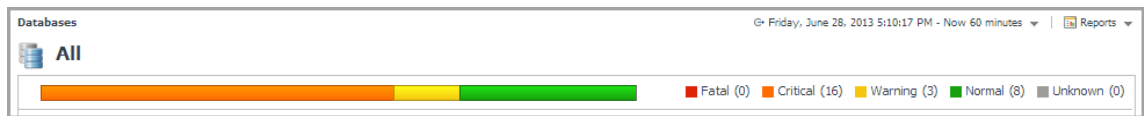
Database Environment Tiles

The Database Environment tiles display a high-level overview of your database environment. There is a tile for each type of database currently monitored by Foglight: **All Instances**, **SQL Server**, **Oracle**, **Sybase**, and **DB2**. Each tile shows how many of the corresponding object instances there are in your database infrastructure, and the count of objects of that type in each of the alarm states (Normal, Warning, Critical, Fatal).

Clicking the object type icon, the object type name, or the object count, shows summary information for that object type in the Databases Instances View. Clicking an alarm state (for example, Warning) on a tile displays summary information in the Databases Instances View for the objects in the selected state.

Figure 2. Database Environment tiles

Description of the Pane

Data displayed	• **Alarm counts.** The total counts of alarms associated with the database instances, broken down by alarm types (Normal, Warning, Critical, Fatal).
	• **Database instance count.** The number of the DB2 database instances in your environment.
Where to go next	Drill down on:
	• **Alarm counts.** Lists the database instances against these alarms generated in the Databases Instances View.
	• **Database instance count.** Displays all DB2 database instances in the Databases Instances View.

Databases Instances View

The Database Instances view displays a list of database instances. It displays all available database instances, or a filtered list. For more information about possible filters, see Instance home page on page 45.

Figure 3. Database Instance view
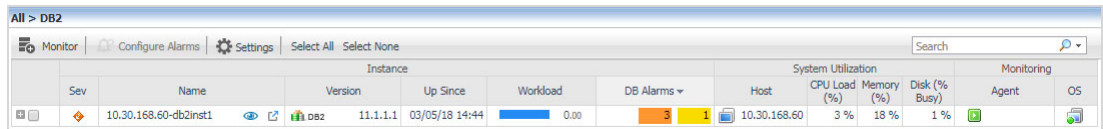
Description of the Pane

Data displayed	• **Database, Sev.** The DB2 database instance's highest severity alarm, used in Foglight to determine the instance's health state.

• **Database, Name, Version, and Up Since.** The name and version number of the database instance, and the date and time when the database instance was last restarted.

NOTE: The Version column does not display any information for DB2 instances running on Windows hosts.

NOTE: If the Name column contains the note `Upgrade required`, the agent package is not re-deployed to the Foglight Agent Manager instance following the cartridge upgrade.

• **Database, Workload.** The workload graph for the DB2 database instance.

• **Database, Alarms.** The number of Warning, Critical, and Fatal alarms for the DB2 database instance. The alarms are displayed by their severity levels, with the aggregated number for each severity.

• **System Utilization, Host.** The name of the computer on which the DB2 database instance is running.

• **System Utilization, CPU Load (%).** The overall operating system CPU usage.

• **System Utilization, Memory (%).** The amount of memory consumed by all OS processes (including the DB2 instance process). It includes both RAM resident memory and swapped memory.

• **System Utilization, Disk (% Busy).** The percentage of time the busiest device spent serving system-wide I/O requests. This metric serves as a measure for the system I/O load.

• **Monitoring Status, Agent.** The operational status of the monitoring agent. When the agent instance is running, hovering over the icon in the State column displays the status message `Collecting Data`.

When the agent instance is running but not collecting data, hovering over the State icon displays one of the following status messages:

- `Starting`
- `Stopped`
- `Stopping`
- `Unknown`
- `Failed to start data collection`

• **Monitoring Status, OS.** Indicates whether Foglight *for DB2 LUW* is monitoring the instance's operating system.

Where to go next	Drill down on:

Description of the Pane

Database Alarms. Display the Outstanding Alarms dialog box.
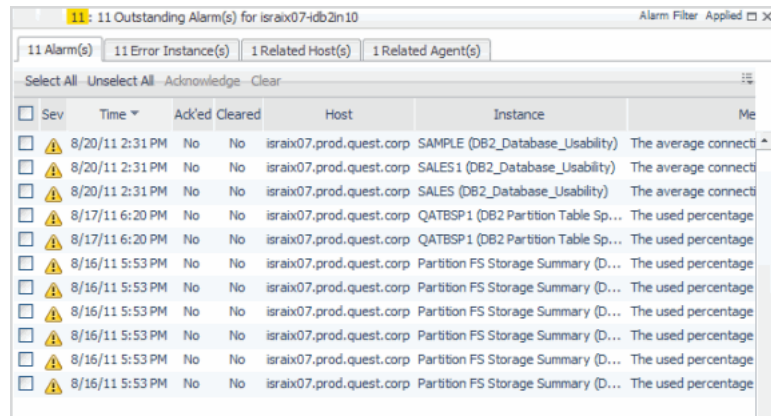
The pop-up shows the most recent alarms generated for the specific database instance. To view additional information about an alarm, click its entry in the Sev column. The Alarm dialog box appears, showing details about the selected alarm.

For complete information about alarms in Foglight, see the *Foglight User Help*.

- To drill down to the Instance home page, click the instance name or click the icon and select **Overview** from the context menu. For more information, see Instance home page on page 45.
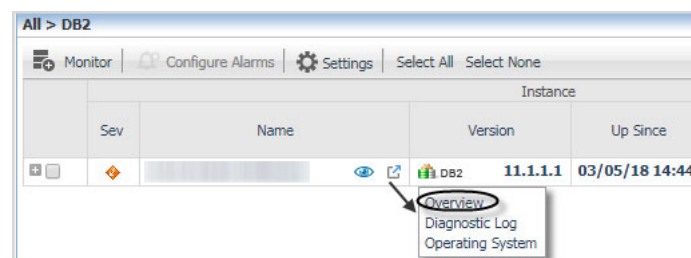
- To display the cue card for an instance, click the (Show quick view) icon.
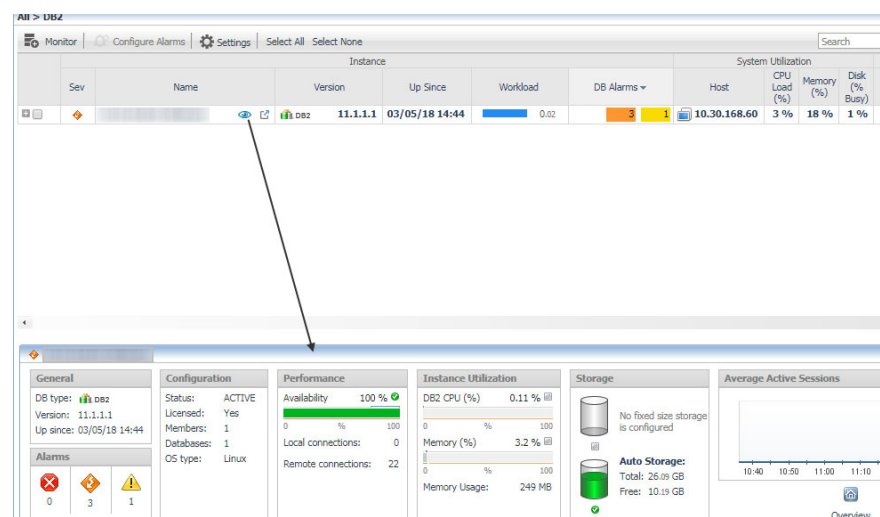
Database Cue Card

The Database Cue Card displays details about a database instance selected in the Databases Instances View.

Figure 4. Database Cue Card
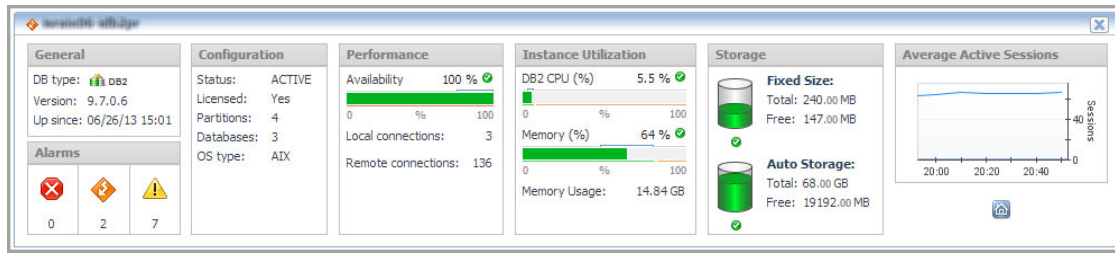
Description of the Pane

Data displayed	
	• **Alarms,** ⬥. The number of Critical alarms associated with the selected database instance.
	• **Alarms,** ⚠. The number of Warning alarms associated with the selected database instance.
	• **Alarms,** ⊗. The numbers of Fatal alarms associated with the selected database instance.
	• **General, DB type.** Specifies the type of the monitored database instance: **DB2**.
	• **General, Version.** Version number of the DB2 database.
	• **General, Up since.** How long the instance has been up and running.
	• **Configuration, Status.** If the instance is configured or not.
	• **Configuration, Licensed.** If the instance is licensed or not.
	• **Configuration, Members.** Number of members in the instance.
	• **Configuration, Databases.** Number of databases in the instance.
	• **Configuration, OS type.** Operating system type.
	• **Performance, Availability.** Availability of the instance in percent.
	• **Performance, Local connections.** Number of local connections.
	• **Performance, Remote connections.** Number of remote connections.
	• **Instance Utilization, DB2 CPU(%).** Percentage of the CPU used by DB2.
	• **Instance Utilization, Memory (%).** Percentage of the memory used by DB2.
	• **Instance Utilization, Memory Usage.** Memory usage in MB.
	• **Storage, Fixed Size.** The total and free amount of fixed storage in MB.
	• **Storage, Auto Storage.** The total and free amount of auto storage in MB.
	• **Average Active Sessions.** The average number of active sessions compared to the time of those sessions.
Where to go next	Clicking the **Overview** icon in the lower right corner takes you to the Instance home page.
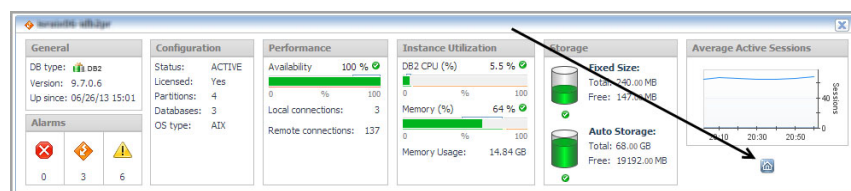
Instance home page

The Instance home page provides metrics on the overall status of the selected instance.
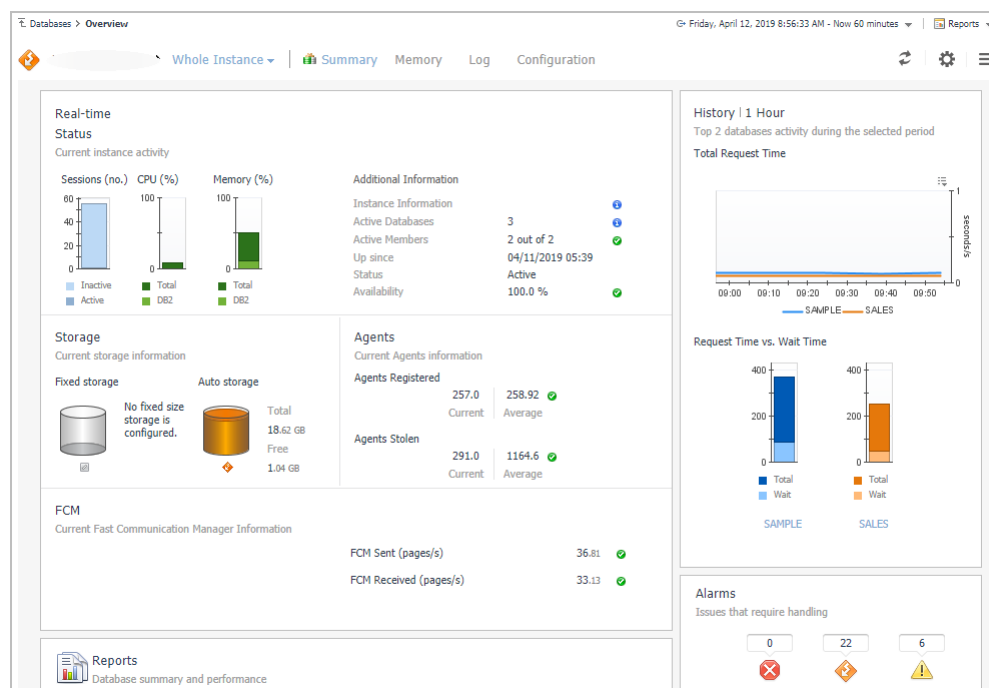
How to Get Here

To open the Instance home page, from the Databases dashboard, find the instance you want to explore and click the instance name (or click and select **Overview**). The Databases > Overview dashboard opens with a summary of the selected instance.

From here, you can drill down to metrics on memory allocation and perform configuration tasks.
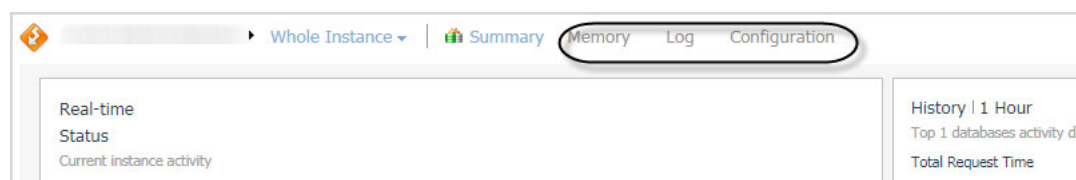
The following summary and drill-downs pages are provided:

- Summary
- Memory Drill-down
- Log Drill-down
- Configuration Drill-down (links to description under Database home page)

Summary

By default, the Summary page view is displayed on the Instance home page. The Summary page contains the following embedded panes:

- Status
- Storage

- Agents
- FCM | pureScale CF
- Reports
- History
- Alarms

Status

The Status pane displays metrics that provide an overview of the current state of the DB2 instance. It is at the top of the Instance home page.

Description of the pane

Data displayed	• **Sessions (no.).** The total number of active and inactive sessions within the instance.
	• **CPU (%).** Total is the average CPU usage of all the underlying hosts on which the instance resides. DB2 is the total CPU usage by all DB2 processes running on the host.
	• **Memory (%).** Total is the average memory usage of all the underlying hosts on which the instance resides. DB2 is the total memory usage by all DB2 processes running on the host.
	• **Instance Information.** The configuration information for the instance (including the product version and whether the product is licensed). Click the icon for the configuration information.
	• **Active Databases.** The total number of local databases within the instance that have applications connected.
	• **Active Members.** The total number of active members within the instance.
	• **Up since.** The date and time the instance was started.
	• **Status.** Whether the instance is active.
	• **Availability.** The current availability of the instance.
	• *pureScale only* — **Active Members**. Number of nodes in the pureScale environment.
	• *pureScale only* — **Active CF**. Displays information about configured CF nodes and their states.
	• *pureScale only* — **pureScale Alerts**. Total number of alerts DB2 reported on the pureScale environment.
Where to go next	• Clicking a data point on any of the three charts displays a pop-up that provides additional, related information. • Clicking the icon next to the Active Databases or Active Members metric displays a pop-up that provides additional information. • Clicking the icon next to the Availability metric displays a pop-up that provides additional, related information.

Storage

The Storage pane provides current storage information for the entire instance. It is at the left of the Instance home page.

Description of the pane

Data displayed	• **Fixed storage.** The total disk space that is available for use by fixed-size tablespaces and the disk space that is still free to be used by fixed-size tablespaces.
	• **Auto storage.** The total file system space that is available for the instance to use to carry out auto storage tablespace operations. Also, the file system space that is still free for the instance to use to carry out auto storage tablespace operations.
Where to go next	Clicking the icon below either of the charts displays a pop-up that provides additional, related information.

Agents

The Agents pane provides registered and stolen agents information for the entire instance. It is located in the middle of the Instance home page.

Description of the pane

Data displayed	• **Agents Registered.** Displays the current agents registered to the instance, and the average agents registered to the instance over the course of the selected time range.
	• **Agents Stolen.** Displays the current agents stolen from the instance, and the average agents stolen from the instance over the course of the selected time range.
Where to go next	Clicking the icon next to either of the metrics displays a pop-up that provides additional, related information.

FCM | pureScale CF

The FCM pane displays FCM metrics for a partitioned instance. It is at the left and in the middle of the Instance home page.

Description of the pane — FCM

Data displayed	• **FCM Sent (page/s).** The total number of FCM buffer pages sent from the current node to the remote node, over the course of the selected time range.
	• **FCM Received (page/s).** The total number of FCM buffer pages received by the current node from the remote node, over the course of the selected time range.
Where to go next	Clicking the icon next to either of the metrics displays a pop-up that provides additional, related information.

When an instance is hosted in a DB2 pureScale environment, you can click the pureScale CF link to see metrics about the cluster cache facility (CF).

Description of the pane — pureScale CF

Data displayed	• **Current Primary ID.** The ID of the CF component.
	• **CPU Usage**. Percentage of overall CPU usage on the CF machine.
	• **Current Host**. The current host of the selected DB2 instance. In a pureScale environment, the host can change depending on load.
	• **Memory Utilization**. Percentage of overall memory usage on the CF component.
Where to go next	Clicking the icon next to either of the metrics displays a pop-up that provides additional, related information.

Reports

The Reports pane provides access to the Executive Workload Summary Report, which is a workload overview of the instance and of each database. It is at the bottom left of the Instance home page.

Click the **Executive Summary Report** link to access the report.

History

The History pane displays the instance activity over the course of the selected time range. It is at the upper right of the Instance home page.

Description of the pane

Data displayed	• **Top Database Workload**. The top three (if available) database workloads, over the course of the selected time range.
	• **Top Database Request Time.** The top three (if available) database total request times and wait request times, over the course of the selected time range.
Where to go next	• Clicking a data point on the "Top Database Workload" chart displays a pop-up that provides more detailed information about the data point.
	• Clicking a data point on a "Top Database Request Time" chart displays a pop-up that provides additional, related information.
	• Clicking the link below any of the "Top Database Request Time" charts takes you to the corresponding Database home page.

Alarms

The Alarms pane displays the total number of fatal, critical, and warning alarms that are currently outstanding against the instance. It is at the lower right of the Instance home page.

Description of the pane

Data displayed	• **Fatal.** The total number of fatal alarms that are currently outstanding against the instance.
	• **Critical.** The total number of critical alarms that are currently outstanding against the instance.
	• **Warning.** The total number of warning alarms that are currently outstanding against the instance.
Where to go next	Clicking either of the three numbers opens the Outstanding Alarm(s) dialog box, which provides a list of the outstanding alarms and a brief description of each.
	Clicking any of the rows in the Outstanding Alarm dialog box opens the Alarm dialog box, which provides detailed information about the associated alarm.

Memory Drill-down

The Memory view provides metrics on how the databases in the instance use memory.
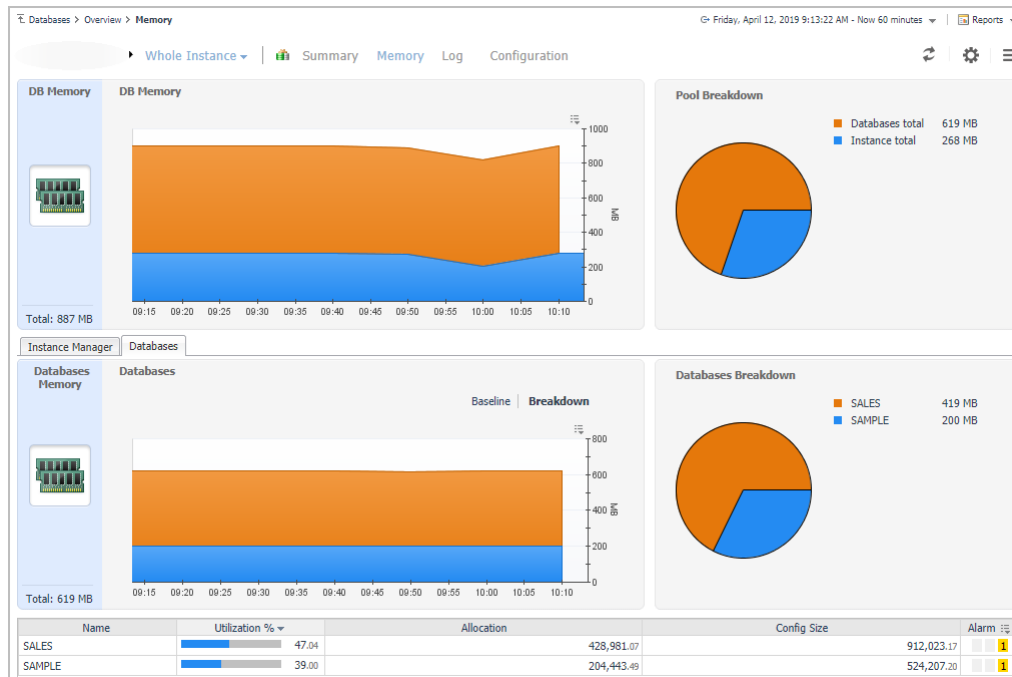
How to Get Here

On the Instance home page, click **Memory**.

This home page contains the following embedded views:

- DB Memory and Pool Breakdown
- Instance Manager tab
- Databases tab
- Primary CF tab (*pureScale only*)
- Backup CF tab (*pureScale only*)

DB Memory and Pool Breakdown

Shows how the total memory available for the instance is allocated to the database and CF pools.

Description of the Pane

Data displayed	• **DB Memory.** Displays the total memory allocated to the selected instance and its databases in a stacking area chart. For pureScale environments, also shows primary and backup CF allocations.
	• **Pool Breakdown.** Displays the total memory allocated to the selected instance and its databases as a pie chart. For pureScale environments, also shows primary and backup CF allocations.

Instance Manager tab

Shows where the instance gets its memory.

Description of the Pane

Data displayed — Charts	• **Instance Manager Memory.** Displays the total memory available to the instance.
	• **Pool Allocated Size.** Displays the pools that contribute to the memory for the instance as a stacking area chart.
	• **Instance Manager Breakdown.** Displays the pools that contribute to the memory for the instance as a pie chart.
Data displayed — Table	• **Pool Name.** The name of the memory pool.
	• **Utilization %.** The percentage of configured allocated memory that is in use.
	• **Allocation.** The average allocated size (in KB) of a memory pool.
	• **Config Size.** The internally configured size (in KB) of a memory pool in the DB2 database system.
	• **Alarm.** The alarm count and state of the memory pool. Click an alarm count to view information about the alarm on the Outstanding Alarms dialog box. For complete information about alarms in Foglight, see the *Foglight User Help*.

Databases tab

Shows memory utilization by database.

Description of the Pane

Data displayed — Charts	• **Databases Memory.** Displays the total memory allocated to the databases in the instance as a stacking area chart.
	• **Databases Breakdown.** Displays the total memory allocated to the databases in the instance as a pie chart.
Data displayed — Table	• **Name.** The name of the databases in the selected instance.
	• **Utilization %.** The percentage of configured allocated memory that is in use.
	• **Allocation.** The average allocated size (in KB) of memory allocated to the database.
	• **Config Size.** The internally configured size (in KB) of memory allocated to the database.
	• **Alarm.** The alarm count and state of the database. Click an alarm count to view information about the alarm on the Outstanding Alarms dialog box. For complete information about alarms in Foglight, see the *Foglight User Help*.

Primary CF tab

Displayed for pureScale environments only. Shows how the CF memory is allocated.

Description of the Pane

Data displayed — Charts	• **Primary CF Memory.** Displays the total cluster cache facility (CF) memory available to the instance and a breakdown of how the cache is allocated for locks, GBP, SCA, and other.
	• **CF Pool Allocation Breakdown.** Displays a breakdown of how the CF pool is allocated for locks, GBP, SCA, and other as a pie chart.
Data displayed — Table	• **Pool Name.** The name of the buffer pools in the CF component.
	• **Utilization %.** The percentage of configured allocated memory that is in use.
	• **Allocation.** The average allocated size (in KB) of a buffer pool.

Description of the Pane

- **Config Size.** The internally configured size (in KB) of a buffer pool in the DB2 database system.

- **Alarm.** The alarm count and state of the buffer pool. Click an alarm count to view information about the alarm on the Outstanding Alarms dialog box. For complete information about alarms in Foglight, see the *Foglight User Help*.

Backup CF tab

Displayed for pureScale environments only. Contains the same information as the Primary CF tab, but the monitoring data represents the backup cluster cache facility.

Database home page

The DB2 Database home page provides information on the status of the entire DB2 database, over the course of the selected time range.
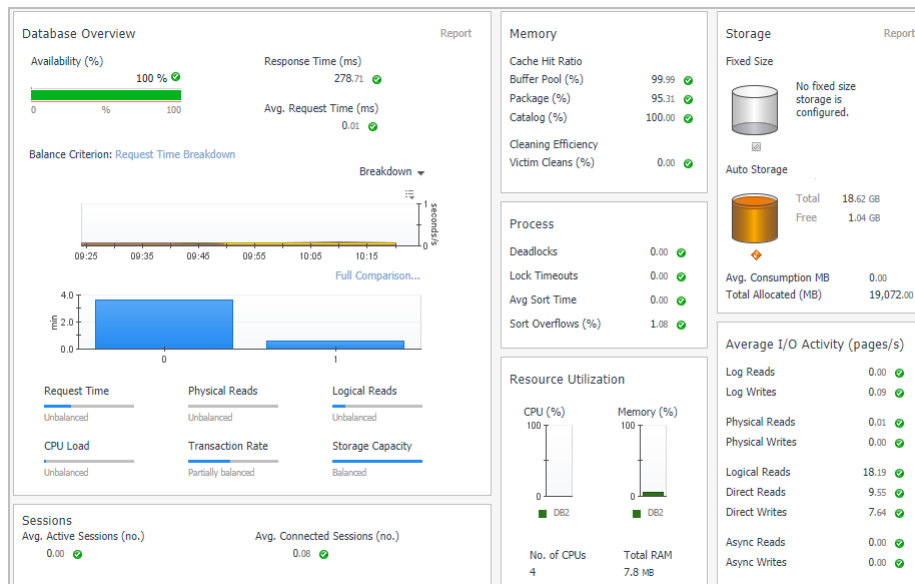
How to Get Here

To open the Database home page, from the Databases dashboard, find the database you want to explore and click the instance name (or click and select **Overview**). The Databases > Overview dashboard opens with a summary of the selected database.

From the Database home page, you can drill down to other pages that display more detailed metrics about the database.
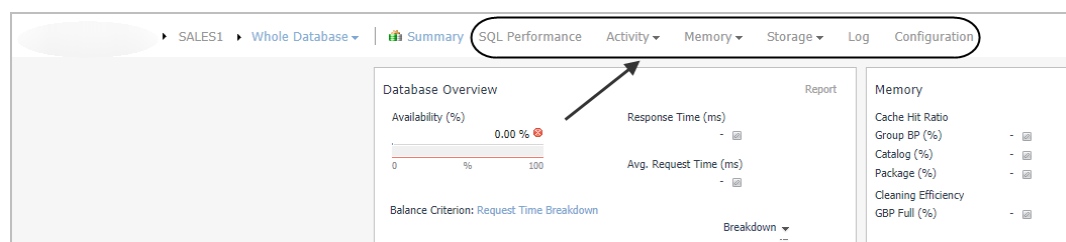
The following summary and drill-downs pages are provided:

Summary

By default, the Summary view is displayed on the Database home page. The Summary contains the following panes:

- Database Overview
- Sessions
- FCM
- Cache Hit Ratio
- Resource Utilization
- Storage
- Average I/O Activity

Database Overview

The Database Overview pane displays metrics that provide an overview of the status of the entire database, over the course of the selected time range. For a partitioned database, this pane also provides a way to examine the balance of certain individual metrics across the members. The pane is at the left of the Database home page.

Description of the pane

Data displayed	
	• **Availability (%).** The availability of the agent's connection FCM.
	• **Response Time (ms).** The average response time of a query when the connection is already open, over the course of the selected time range.
	• **Avg. Request Time (ms).** The average duration for a single database request, over the course of the selected time range. This metric is only available for DB2 v9.7.0.1 and later.
	• **Balance Criterion.** The level of balance exhibited, for a particular database criterion (for example, workload in terms of the number of connections), over the course of the selected time range. Click the arrow at the right of "Balance Criterion" to choose a different criterion from the list.
	• **Workload.** The balance of average active agents across the database members. If most of the activity is on one member, for example, the activity is not balanced. If most of the activity is spread equally across the members, it is balanced.
	• **Request Time.** The balance of requests across the monitored members, or the extent to which the total request time is spread across the members. This metric is only available for DB2 v9.7.0.1 and later.
	• **Physical Reads.** The balance of the physical reads across the monitored members, or the extent to which the physical reads operations are balanced across the members.

Description of the pane

	• **Logical Reads.** The balance of the logical reads across the monitored members, or the extent to which the logical reads operations are balanced across the members.
	• **CPU Load.** The balance of CPU used across the monitored members.
	• **Transaction Rate.** The balance of the transactions across the monitored members. For example, if most of the transactions are done on one member, the transactions are not balanced. If they are spread equally across the members, they are balanced.
	• **Storage Capacity.** The balance of storage capacity used across the monitored members.
Where to go next	Clicking the icon next to the Availability, Response Time, or Avg. Request Time metric displays a pop-up that provides additional, related information.
	Mousing over a data point on either of the Balance Criterion charts displays a pop-up that provides more detailed information about that data point.
	Clicking any of the Request Time, Physical Reads, Logical Reads, CPU Load, Transaction Rate, or Storage Capacity charts displays a pop-up that provides additional, related information.

Sessions

The Sessions pane displays client application session metrics for the entire database. It is at the lower left of the Database home page.

Description of the pane

Data displayed	• **Avg. Active Sessions (no.).** The average number of active sessions connected to the database over the course of the selected time range.
	• **Avg. Connected Sessions (no.).** The average number of sessions connected to the database over the course of the selected time range.
Where to go next	Clicking the icon next to either of the metrics displays a pop-up that provides additional, related information.

FCM

The FCM pane displays FCM metrics for the entire database. It is located in the upper middle of the Database home page.

Description of the pane

Data displayed	• **Total Buffers Sent.** The total number of FCM buffer pages sent from the current node to the remote node, over the course of the selected time range.
	• **Total Buffers Received.** The total number of FCM buffer pages received by the current node from the remote node, over the course of the selected time range.
Where to go next	Clicking the icon next to either of the metrics displays a pop-up that provides additional, related information.

Cache Hit Ratio

The Cache Hit Ratio pane displays cache hit ratio metrics for the entire database. It is located toward the center of the Database home page.

Description of the pane

Data displayed	• **Buffer Pool (%).** The percentage of database page I/O requests satisfied using the buffer cache (and therefore the database did not have to perform disk reads), over the course of the selected time range.
	• **Package (%).** The number of times that a requested section was not available for use and had to be loaded into the package cache. Calculated as a percentage of the total number of times that a section was requested, over the course of the selected time range.
	• **Catalog (%).** The number of times that a table descriptor or authorization information was not found in the catalog cache and had to be inserted ther. Calculated as a percentage of the total number of times that a table descriptor or authorization information was requested, over the course of the selected time range.
	• *pureScale only* — **Overall BP (%)**. A percentage that reflects the number of times a requested page was found in the buffer pool out of all times it was requested. If a page is found in the buffer pool, it saves the database the time to look for it in the physical storage.
	• *pureScale only* — **Global BP (%)**. A percentage that reflects the number of times a requested page was found in the GBP out of all times it was requested. If a page is found in the GBP, it saves the database the time to look for it in the LBP.
	• *pureScale only* — **GBP Full (%)**. A percentage that reflects the number of times a Group Buffer Pool Full condition occurred per 10,000 commits.
Where to go next	• Clicking the icon next to any of the three metrics displays a pop-up that provides additional, related information. • Clicking the link at the bottom of any of the three pop-ups takes you to the corresponding dashboard for additional information.

Resource Utilization

The Resource Utilization pane provides information on CPU and RAM consumption for the entire database. It is located in the lower middle of the Database home page.

Description of the pane

Data displayed	• **CPU (%).** Total CPU usage by all processes (including all DB2 processes) running on the host, as a percentage of the total CPU available.
	• **No. of CPUs.** The total number of CPUs available.
	• **Memory (%).** Total memory usage by all processes (including all DB2 processes) running on the host, as a percentage of the total memory available.
	• **Total RAM.** The total RAM (in MB) available.
Where to go next	Mousing over a data point on either of the CPU or Memory charts displays a pop-up that provides more detailed information about the data point. Clicking a data point on either of the CPU or Memory charts displays a pop-up that provides additional, related information.

Storage

The Storage pane provides information about fixed and auto storage for the entire database. It is at the upper right of the Database home page.

Description of the pane

Data displayed	• **Fixed Size (MB).** The total disk space that is available for use by fixed-size tablespaces and the disk space that is still free to be used by fixed-size tablespaces.
	• **Auto Storage (MB).** The total file system space that is available for the database to use to carry out auto storage tablespace operations, and the file system space that is still free for the database to use to carry out auto storage tablespace operations.
Where to go next	• Clicking the icon below either chart displays a pop-up that provides additional, related information.
	• Clicking the **Investigate using the "Tablespaces" drilldown** link at the bottom of either pop-up takes you to the Tablespaces dashboard for additional information.

Average I/O Activity

The Average I/O Activity pane provides I/O metrics for the entire database. It is at the lower right of the Database home page.

Description of the pane

Data displayed	• **Log Reads.** The number of times the log was read from during the selected time range.
	• **Log Writes.** The number of times the log was written to during the selected time range.
	• **Physical Reads.** The number of reads directly from disk during the selected time range.
	• **Physical Writes.** The number of writes directly to disk during the selected time range.
	• **Logical Reads.** The number of reads from the buffer pool during the selected time range.
	• **Direct Reads.** The number of reads that did not use the buffer pool, during the selected time range.
	• **Direct Writes.** The number of writes that did not use the buffer pool, during the selected time range.
	• **Async Reads.** The number of asynchronous reads during the selected time range.
	• **Async Writes.** The number of asynchronous writes during the selected time range.
Where to go next	Clicking the icon next to any of the metrics displays a pop-up that provides additional, related information.

SQL Performance

SQL Performance page provides the ability to investigate the activity and resource consumption of a selected database or database partition.

The SQL Performance page displays the following components:

- Database (or Partition) View
- SQL Statements View
- Single Statement View

Selecting different levels in the History Tree to display different views in the History section view. For details, see Database (or Partition) View, SQL Statements View, or Single Statement View.

> **NOTE:**
> The SQL Performance dashboard only support DB2 version no lower than v9.7.0.10. If the version is below v9.7.0.10, no data will be displayed in the *SQL Statements* view and the *Single Statements* view.
>
> After the upgrade, historical Top SQL data is available in the topology but not accessible by the UI.

Database (or Partition) View

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

- **Resource consumption charts** — This section displays data in four different charts:

  - **Workload chart** — Displays the database (or Partition) resource activity over the selected time frame by emphasizing the resources by colors.

    - **Baseline chart** — Displays the database (or Partition) workload compared to the baseline over time.
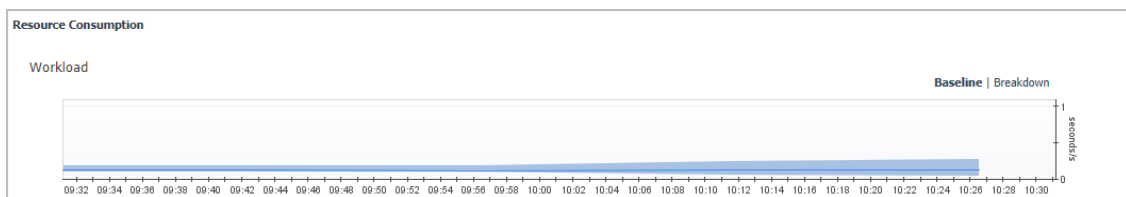
    - **Breakdown chart** — Rate of activity of the database (or Partition) per second.
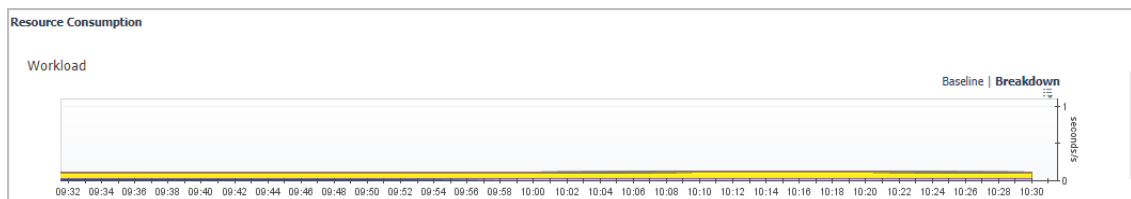
  - **Resource Breakdown Pie chart** — Displays the resource breakdown usage by % of the total database (or Partition) activity.
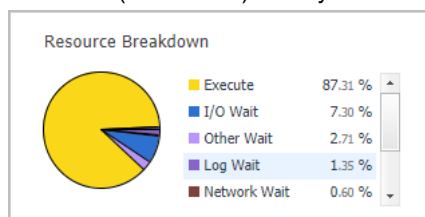
- **Metrics Overview** - Displays a graphical representation of the metrics highlighted in the Workload related Metrics table below.
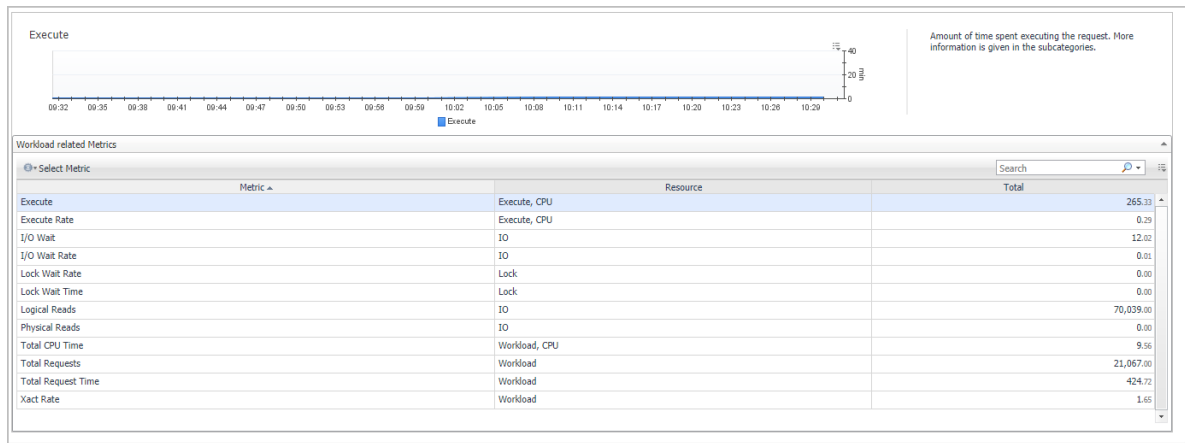
- **Workload related Metrics** - A table that displays a variety of resource consumption metrics which can give an in-depth of the database/partition activity. Metrics are categorized by resource and the resource name can be used to filter the table by writing its name in the search box.

The table below lists the Workload related metrics and their descriptions:

Name	Descriptions
Total Requests	The total amount of requests completed.
Total Requests Time	The total amount of time spent working on requests. This value is reported in Seconds.
Total Requests Time Rate	The rate of time spent working on requests per second. This value is reported in Seconds/s.
Total CPU Time	The total amount of CPU time used while within DB2®. Represents total of both user and system CPU time. This value is reported in Seconds.
Total CPU Wait	The total time that requests, that were run in this service class, spent waiting to access the CPU. This value is given in seconds. Available for version 10 and above.
FCM Receive Volume	The total amount of data received via the FCM communications layer. This value is reported in bytes.
FCM Send Volume	The total amount of data distributed by the FCM communications layer. This value is reported in bytes.
FCM Receive Volume Rate	The rate of data received via the FCM communications layer per second. This value is reported in bytes/s.
FCM Send Volume Rate	The rate of data distributed by the FCM communications layer per second. This value is reported in bytes/s.
TCP/IP Send Volume	The amount of data sent by data server to client. This value is reported in kbytes.
TCP/IP Send Volume Rate	Number of kbytes sent by the data server to clients per second.
TCP/IP Receive Volume	The amount of data received by the data server from clients over TCP/IP. This value is reported in kbytes.
TCP/IP Receive Volume Rate	Number of kbytes received by the server from clients per second.
Physical Reads	Total physical reads (index, XDA and data) in the bufferpool.

Name	Descriptions
Logical Reads	Total logical reads (index, XDA and data) in the bufferpool.
Xact Rate	The number, per second, of transactions that were carried out during the specified time range.
Lock Waits	The number of times that applications or connections waited for locks.
Lock Escalation	The number of times that locks have been escalated from several row locks to a table lock.
Lock Timeouts	The number of times that a request to lock an object timed-out instead of being granted.
Deadlocks	A deadlock occurs when there is a cyclic dependency between two or more threads, or processes, which contend for the same set of resources within DB2, namely: each task has a lock on a resource which the other tasks are trying to lock.
I/O Wait	Amount of time spent waiting for I/O. Subcategories specify further information.
I/O Wait Rate	The rate of total time spent waiting for I/O per second.
Lock Wait Time	Amount of time spent waiting for locks.
Lock Wait Rate	The rate of total time spent waiting for locks per second.
FCM Wait	Amount of time spent waiting for the FCM communications layer.
FCM Wait Rate	The rate of total time spent waiting for the FCM communications layer per second.
Network Wait	Amount of time spent waiting on network protocols. More information is given in the subcategories.
Network Wait Rate	The rate of total time spent waiting on network protocols per second.
Log Wait	Amount of time spent waiting on the log. More information is given in the subcategories.
Log Wait Rate	The rate of total time spent waiting on the log per second.
pureScale Wait	Amount of time spent waiting for various functions within the cluster. More information is given in the subcategories.
pureScale Wait Rate	The rate of total time spent waiting for various functions within the cluster per second.
Memory Wait	Amount of time spent waiting on an internal latch.
Memory Wait Rate	The rate of total time spent waiting on an internal latch per second.
Other Wait	Aggregate several events not usually with high impact on performance. More information is given in the subcategories.
Other Wait Rate	The rate of total time spent aggregating several events not usually with high impact on performance per second.
Execute Time	Total time spent executing routines and performing section execution. Processing time does not include wait time.
Execute Time Rate	The rate of total time spent executing routines and performing section execution per second.
Compile	Amount of time spent doing explicit or implicit compiles.

Name	Descriptions
Compile Rate	The rate of total time spent doing explicit or implicit compiles per second.
Transaction	Amount of time spent on commit or rollback of transaction.
Transaction Rate	The rate of total time spent on commit or rollback of transaction spent per second.
Administration Time	Amount of time spent processing administration tasks (index build, backup, load and reorg).
Administration Time Rate	The rate of total time spent on processing administration tasks' spent per second.

> **i** | **NOTE:** If the monitored DB2 server is v9.7.0.10 or v9.7.0.11, the following metrics are not available: Memory Wait, Memory Wait Rate, pureScale Wait, pureScale Wait Rate, and Total CPU Wait.

SQL Statements View

Resource consumption charts

For details, refer to Resource consumption charts under Database (or Partition) View.
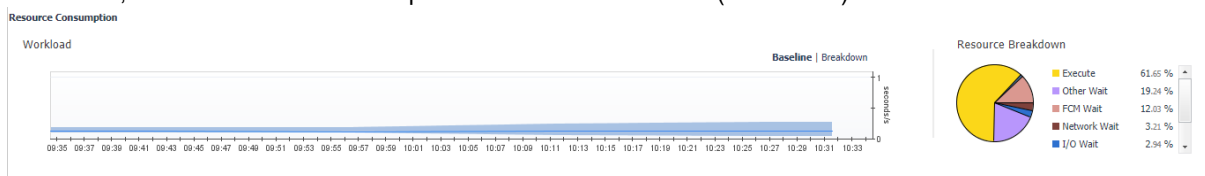
Top SQL Statement table

The Top SQL Statements table displays the top SQL statements by their overall active time during the selected time range. The number of SQL statements displayed is shown at the table's title. Selecting a row displays the CPU workload and executions of the selected SQL statement on the SQL Activity section below. For details, see SQL Activity section. Selecting the SQL Text in the SQL Statements tree panel displays a page that allows viewing detailed information about the selected SQL statement.

The table below lists the Top SQL Statements name and their descriptions:

Name	Description
SQL Text	The SQL text for the selected SQL statement.
Executions	The number of times that an SQL statement has been executed.
Total Execution Time (seconds)	The total time in seconds that was spent executing the particular statement. Not including time spent executing routines used as part of the statement.
Avg Response Time (seconds)	The average time in seconds that was spent executing the particular statement Not including time spent executing routines used as part of the statement.
CPU Time (seconds)	The total CPU time consumed by the statement in the selected timeframe.
Rows Read	The number of rows read from the table. This element helps you identify tables with heavy usage for which you may want to create additional indexes.
Rows Modified	The number of rows changed (inserted, deleted or updated) by the statement. A high value for table-level information indicates there is heavy usage of the table and you may want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table.

Name	Description
Rows Returned	The number of rows that have been selected and returned to the application. If this number is significantly lower than "Rows read", then it might indicate an index is missing.
Wait Time (%)	Percent of time spent waiting within the DB2® database server out of the total time spent actively working on requests.
Total Dispatcher Queue Wait (seconds)	The total time that requests, that were run in a service class, spent waiting to access the CPU.
I/O Wait (seconds)	Total time spent waiting, while processing an I/O request. Include read and write operations, both direct and from memory pools.
Execute (seconds)	Total time spent executing routines and performing section execution. Processing time does not include wait time.
Log Wait (seconds)	Total time spent waiting for log records to be flushed to disk or waiting for space in the log buffer.
FCM Wait (seconds)	The time spent waiting for an FCM message to be sent or FCM reply to be received.
Max Query Cost Estimate	The maximum estimated cost of a query out of all cost estimations collected for the query in the timeframe selected. Query cost estimation is determined by the SQL compiler. This value is reported in timerons.
Lock Wait (seconds)	The time spent waiting for locks.
Total Sort Time (seconds)	Total amount of time spent performing sorts while executing a section, which is the execution of the compiled query plan generated by the SQL statement.
Other Wait (seconds)	Total time spent waiting for miscellaneous events such as audit records writes, dialog writes, event monitor records and send or receive data from an in-database analytics process.
WLM Wait (seconds)	The time spent waiting on a WLM queuing threshold.
Compile Time (seconds)	The overall amount of time that was required to prepare the specific SQL statement in the time frame selected.
Memory Wait (seconds)	The amount of time, spent in extended latch waits.
purescale Wait (seconds)	In a DB2® pureScale® environment, this is the time spent waiting for communicating with the cluster caching facility or waiting for page locks, where the lock request caused a page to be reclaimed either locally or from another member.

i | **NOTE:** If the monitored DB2 server is v9.7.0.10 or v9.7.0.11, the following metrics are not available: Memory Wait (seconds) and pureScale Wait (seconds).

SQL Activity

Located below the table of Top SQL Statements, the SQL Activity section of the pane displays the CPU workload and executions of the SQL statement selected, compared to the overall workload and CPU used by all statements in the database/partition.
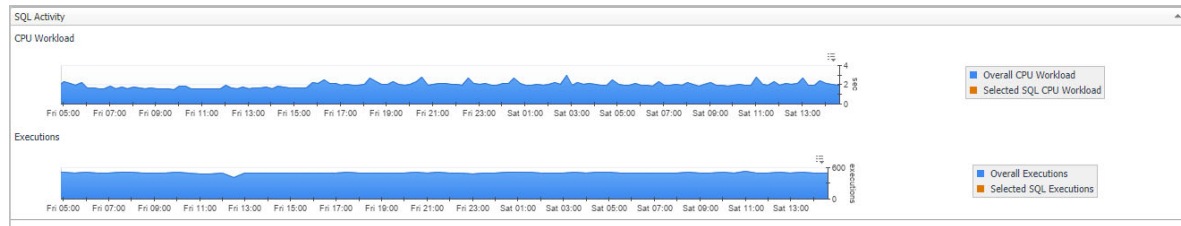
These metrics are visually represented in two graphs:

- **CPU Workload** — Displays the activity (system workload), plotted over the specified time range. This graph compares the following metrics:

    - **Overall CPU Workload** — the CPU usage incurred by the total SQL statements run in the database/partition level.

    - **Selected SQL CPU Workload** — the CPU usage incurred by the selected SQL statement.

- **Executions** — Displays the number of times the selected SQL statement and all of the database (or partition) executed SQL statement in the specified time range. This graph compares the following metrics:

    - **Overall Executions** – The number of the total SQL statements run in the database/partition level.

    - **Selected SQL Executions** - The execution number of the selected SQL statements in the Top SQL statements table.

Single Statement View

To view detailed graphic and textual data about a specific SQL statement, select the SQL statement from the statements listed in the History Tree. This page contains the following panes:

SQL Activity

The SQL Activity pane displays the CPU workload and executions of the selected SQL statement, as well as all SQL statements displayed in the History Tree on the page left side. This pane is identical to the SQL Activity section in the SQL Statements Page.

SQL Text

The SQL Text pane displays an overview of the selected SQL statement text. In addition, the pane includes the following button:

- **View Full Text** - Displays a pop-up with the SQL Statement text, whose maximum size is by default 2048 characters.
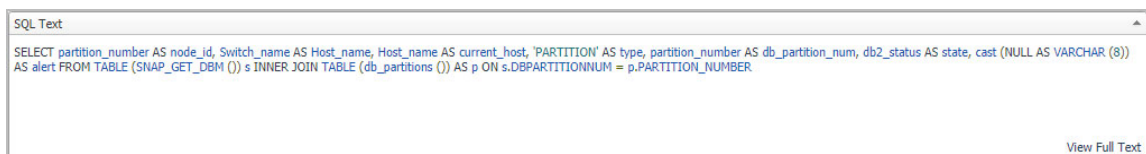
SQL Metrics

The SQL Metrics table displays details about the selected SQL statement.

> **i** | **NOTE:** Switching between Total and Per Execution allows you to see the totals for all executions versus the average execution metrics.

The table below lists the SQL metrics name and their descriptions:

Name	Description
Total Execution Time (seconds)	The total time in seconds that was spent executing the particular statement. Not including time spent executing routines used as part of the statement.
Deadlocks	The number of times that a request to lock an object resulted in a deadlock.
Sort Overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
Rows Read	The number of rows read from the table. This element helps you identify tables with heavy usage for which you may want to create additional indexes.
Rows Modified	The number of rows changed (inserted, deleted or updated) by the statement. A high value for table-level information indicates there is heavy usage of the table and you may want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table.
Rows Returned	The number of rows that have been selected and returned to the application. If this number is significantly lower than "Rows read", then it might indicate an index is missing.
Total CPU Time (seconds)	The total CPU time consumed by the statement in the selected timeframe.
Total Wait Time (seconds)	The total time spent waiting within the DB2® database server out of the total time spent actively working on requests.
Execute (seconds)	The number of times that an SQL statement has been executed.
FCM Wait (seconds)	The time spent waiting for an FCM message to be sent or FCM reply to be received.
I/O Wait (seconds)	Total time spent waiting, while processing an I/O request. Include read and write operations, both direct and from memory pools.
Memory Wait (seconds)	The amount of time, spent in extended latch waits.
Lock Wait (seconds)	The time spent waiting for locks.
Log Wait (seconds)	Total time spent waiting for log records to be flushed to disk or waiting for space in the log buffer.
pureScale Wait (seconds)	In a DB2® pureScale® environment, this is the time spent waiting for communicating with the cluster caching facility or waiting for page locks, where the lock request caused a page to be reclaimed either locally or from another member.
Other Wait (seconds)	Total time spent waiting for miscellaneous events such as audit records writes, dialog writes, event monitor records and send or receive data from an in-database analytics process.
Total Sort Time (seconds)	Total amount of time spent performing sorts while executing a section, which is the execution of the compiled query plan generated by the SQL statement.
Compile Time (seconds)	The overall amount of time that was required to prepare the specific SQL statement in the time frame selected.
Executions	The number of times that an SQL statement has been executed.
Sorts	The total number of times that a set of data was sorted in order to process the statement operation.
Avg Response Time (seconds)	The average response time of a single statement execution.

Pie Charts

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

- **Total CPU Time** — The total CPU time consumed for executing the selected SQL statement, compared with the CPU time consumed for executing all of the SQL statements.
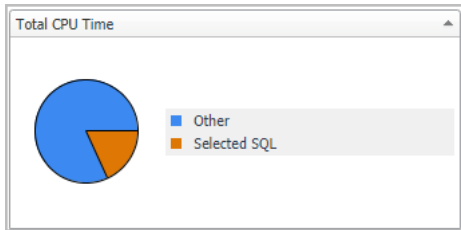
- **Sort Time** —The total time spent on the selected SQL statement's data sorts, compared with the sort time spent to carry out other database processes.
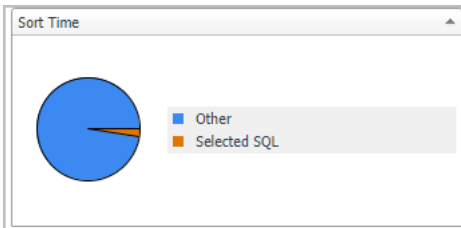
- **Logical Reads** —The total number of logical reads for the selected SQL statement, compared with the total SQL statements.
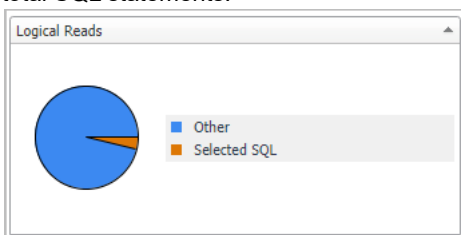
- **Physical Reads** —The total number of physical reads for the selected SQL statement, compared with the total SQL statements.
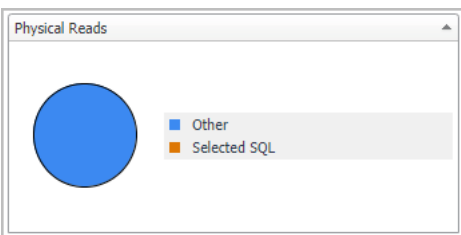
Activity Drill-downs

On the **Database home page** toolbar, click Activity and select one of the following views:

- Top Tables
- Locks
- Current Agents
- User-defined Collections

Top Tables

Provides information about the performance of the tables in the selected database.

By default, the Top Tables pane lists performance metrics for the top 10 tables of the selected database in order of rows read. You can configure the pane to list the top 10, 15, or 20 tables listed in order of rows read, rows written, overflows, or page reorganizations.

Select tables from the list to compare their selected activity metrics on the charts at the bottom of the pane.
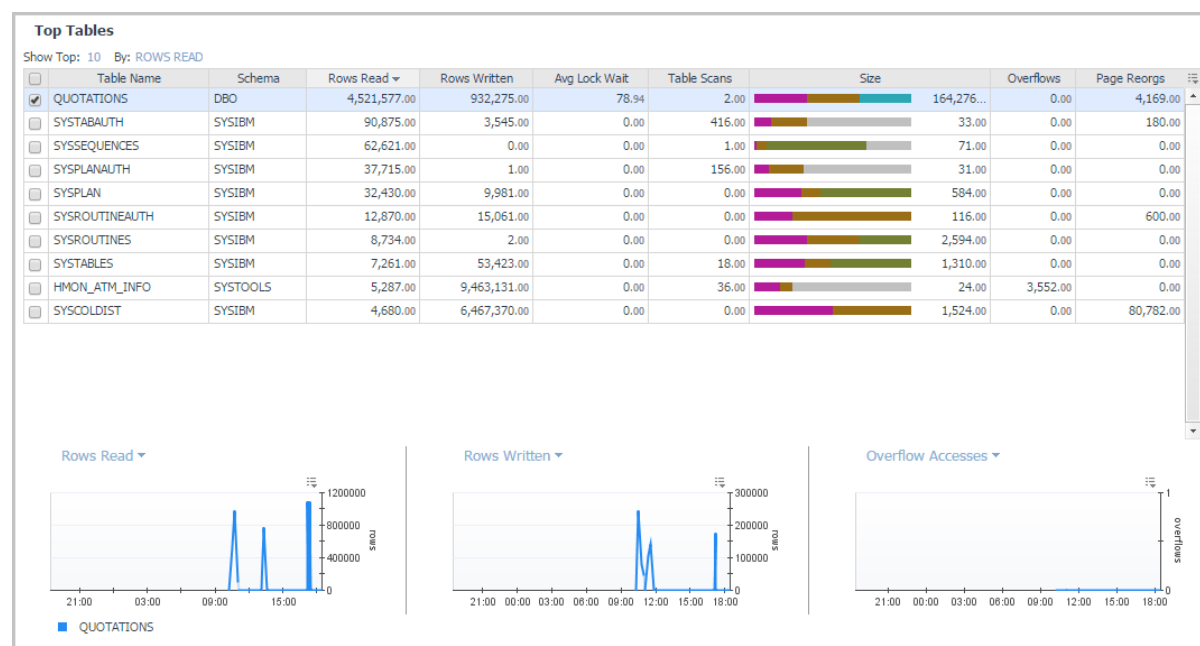
How to Get Here

On the **Database home page** toolbar, click **Activity** and select **Top Tables**.

Description of pane

Data displayed — Top Tables List	
	• **Table Name.** The name of the monitored table.
	• **Schema.** The schema of the monitored table. Used with Table Name, this element can help in determining the source of a contention for resources.
	• **Table Type.** The type of table for which information is returned.
	• **Rows Read.** The number of rows in the table that are read from in order to process the selected SQL statement.
	• **Table organization.** Organization method of the data in the table. 'R' indicates data is row-organized, 'C' indicates column-organized.
	• **Number of columns**. Number of columns defined for the table as appears in the catalog.
	• **Rows Written.** The number of rows in the table that are changed (inserted, updated, or deleted) in order to process the selected SQL statement.
	• **Overflows.** The total number of accesses (reads and writes) to overflowed rows of the table.
	• **Avg Lock wait** - The average wait time for a lock (in seconds).
	• Available from v10.1 and above when the mon_obj_metrics is set to extended.
	• **Avg Num Column Referenced** - Average number of column referenced in each access to the table. Available from v10.5 and above when the mon_obj_metrics parameter is set to extended.

Description of pane

- **Table scans** - The number of scans performed on the table in the selected time frame. Available from version 9.7 and above.

- **Size** - Total size of table in pages separated to the various page types: Data, Index, Xda, Lob, Long and Col

- **Page Reorgs.** The number of page reorganizations run for the table during the specified time range.

- **Tablespace Id.** The integer used by the selected database to uniquely represent this tablespace.

- **Data.** The total number of disk pages the table consumes. This number represents the base table size only. Space consumed by index objects, LOB data, and long data is displayed in the Index, Lob, and Long columns, respectively.

- **Index.** The total number of disk pages consumed by all indices defined in the table.

- **Lob.** The total number of disk pages consumed by LOB data in the table.

- **Long.** The total number of disk pages consumed by LONG data in the table.

- **Xda.** The total number of disk pages consumed by XML storage object (XDA) data in the table.

- Col - The total number of disk pages consumed by column-organized object data (relevant from version 10.5 and above)

- Lock Escals - The number of times that locks have been escalated from several row locks to a table lock. Available from v10.1 and above when the mon_obj_metrics is set to extended

- Lock Escals Global - Number of lock waits due to the application holding the lock being on a remote member. Available from v10.1 and above when the mon_obj_metrics is set to extended. Relevant for pureScale instance only

- Lock waits - The total number of times that applications or connections waited for locks. Available from v10.1 and above when the mon_obj_metrics is set to extended.

- Lock wait time Global - Number of lock waits due to the application holding the lock being on a remote member. Available from v10.1 and above when the mon_obj_metrics is set to extended. Relevant for pureScale instance only

Data Displayed — Top Tables Charts	Displays the same metrics in chart form.

Locks

Provides current and historical metrics on locks held in the selected database.

How to Get Here

On the **Database home page** toolbar, click **Activity** and select **Locks**.

You can also access the Locks pane for a database from the Databases Dashboard. Click the icon at the far right of the database name cell and selecting **Locks** from the menu that appears.

This view provides the following embedded views:

- Locks Analysis — Overall Lock Time
- Locks Analysis — pureScale Lock Time
- Current Locks tab
- Historical Locks tab

Locks Analysis — Overall Lock Time

Displays a selection of lock metrics from the Current Locks Wait Collection.
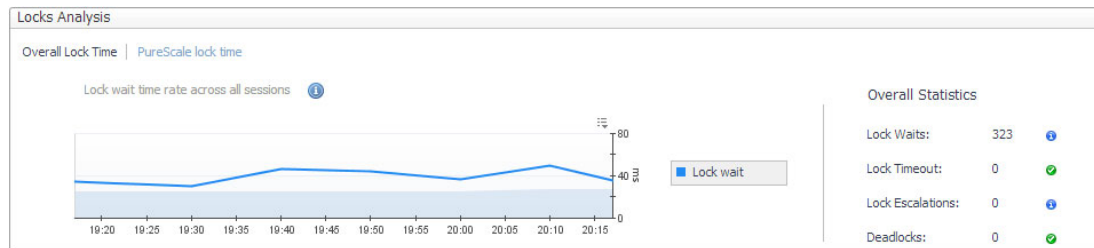
Locks Analysis — pureScale Lock Time

Displays the global lock metrics from the CF Locks Collection.
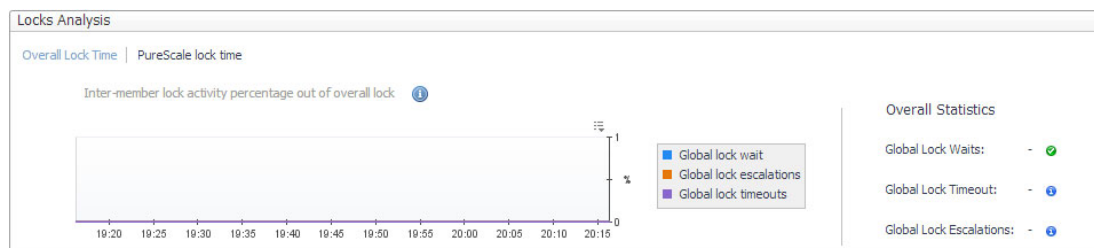
Current Locks tab

Description of the Pane

Data displayed	
	• **Agent ID.** 16-bit counter of a system-wide unique ID for the locked wait application.
	• **Member Num.** The database member from which the data was retrieved for this row.
	• **Lock Mode.** Indicates the lock type being held and is used to determine what is the source of the lock contention.
	• **Lock Object Type.** The object type held by the application that helps determine resource contention.
	• **Lock Wait Start Time.** Date and time the lock started waiting to receive a lock, that had another application lock it.
	• **Lock Duration.** The length of time for the lock wait.
	• **Statement Text**. The full statement text of the locked agent.
	• **Table Name.** Table name.
	• **Table Schema.** Table schema.
	• **Tablespace Name.** Name of the tablespace.
	• **Appl ID Holding Lock.** Application ID of the application holding a lock on the object that the application waits to obtain.
	• **Lock Mode Requested.** Mode of the lock requested by the application.
	• **Lock Attributes.** The attributes of the lock. The text identifier of Y is used if there are no locks.
	• **Lock Name.** Provides a name for the lock.
	• **Lock Escalation.** Indicates whether a lock request was made as part of a lock escalation.
	• **Lock Current Mode.** Lock type before the conversion is completed.
	• **Lock Release Flags.** A flag based on bits from sqlmon.h.

Description of the Pane

- **Subsection Number.** This number relates to the subsection number in the access plan that can be obtained with db2expln.

- **Data Partition ID.** Identifies the data partition for returned information and is only used by partitioned tables. The return information represents a value of -1.

Historical Locks tab

Description of the Pane

Data displayed	• **Longest Duration.** Over the course of the selected time range, the locks with the longest wait durations, displayed at regular intervals.
	• **Number of Locks.** Over the course of the selected time range, the number of locks that exceed the configured lock wait duration threshold, displayed at regular intervals.
	• **Snapshots List.** Select any of the intervals in either of the Longest Duration or Number of Locks charts. The Snapshots List at the bottom displays the snapshots that make up the selected interval.
Where to go next	• Clicking a specific snapshot displays a pop-up of the corresponding Lock Tree.
	For descriptions of the columns in the Lock Tree, see the table in Locks on page 65.

Current Agents

The Current Agents pane displays agent workload, agent activity, and CPU usage.
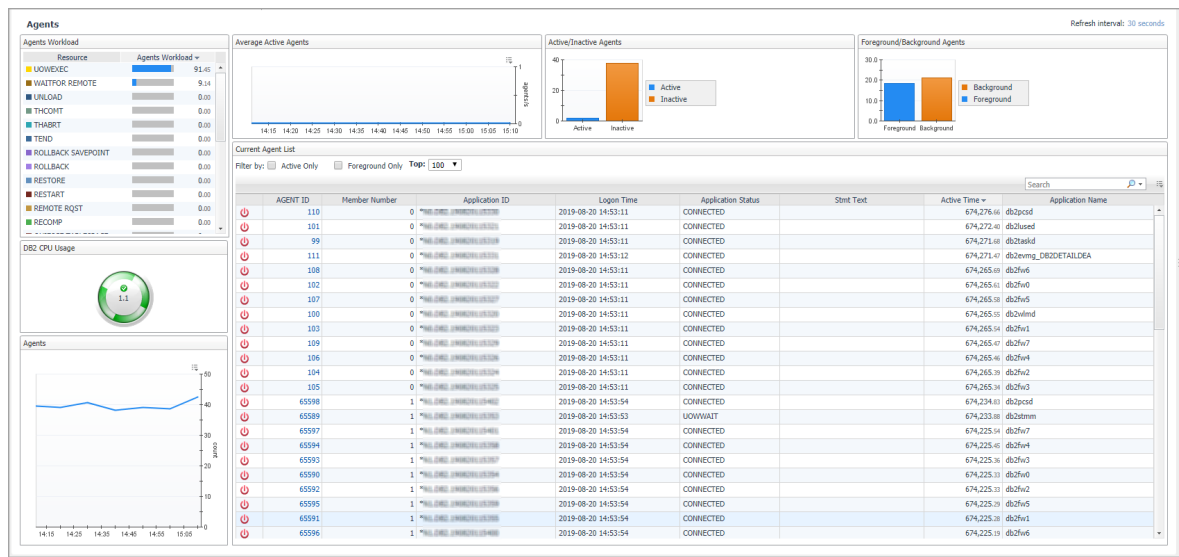
How to Get Here

On the **Database home page** toolbar, click **Activity** and select **Current Agents**.

You can also access the Current Agents pane for a database from the Databases Dashboard. Click the icon at the far right of the database name cell and selecting **Current Agents** from the menu.

This view provides the following embedded views:

- Agents Workload

- Average Active Agents

Agents Workload

Data displayed	**Resource.** Lists the agent workload resources.
	Agent Workload. Displays how much the listed resource uses the agent.
	BACKUP. Indicates the database is being backed up.
	COMMIT. Indicates a work unit is committing its database changes.
	COMP. Reports that the database manager is doing a compilation of a SQL statement or planning a precompiling a plan for the application.
	CONNECTPEND. Reports that the application has started to connect to the database and the connection request is not finished.
	CREATE_DB. Tells the user that a database is being created. An agent initiation request has started to create the database, but the request is not complete.
	DECOUPLED. The agent has been decoupled and no agents are associated with the application running. This state is normal. If the connection is enabled, no dedicated agents exist. Applications can be decoupled on the coordinating partition. In environments that are non-concentrated, applications are not decoupled and a dedicated coordinator agent is always running.
	DISCONNECTEDPEND. Tells the user that the application is in the process of disconnecting, however a command has not finished execution. An application might have begun the disconnection command. The database manager must disconnect from the database if an application finishes and no disconnection has taken place.
	INTR. Tells the user that an interruption of a request is taking place.
	LOAD. Tells the user that an application is doing an expedited loading of data to a database.
	LOCKWAIT. Data being processed is standing by for a lock. When the lock is received, its status goes back to its previous value.
	QUIESCE TABLESPACE. A quiesce tablespace request is taking place.
	RECOMP. A database manager is recompiling or rebinding the application.
	REMOTERQST. A federating data source is in the process of providing results to the application.
	RESTART. After a crash, the database is starting back-up and performing a recovery.
	RESTORE. A backup image of the database is being restored.
	ROLLBACK. The database is rolling back changes.
	ROLLBACK SAVEPOINT. The database is rolling back changes to a certain saving point in the database.

TEND. Work that belongs to a global transaction has ended. This work has not committed to the preparation phase required by the two-step protocol commitment.

THABRT. This work belongs to a global transaction rolled-back based on speculation.

THCOMT. This work belongs to a global transaction committed, based on speculation.

UOWEXEC. Work is being processed by the database manager upon request.

UNLOAD. The database is doing an accelerated unloading of data.

UOWEXEC. Requests are being processed by the database manager.

UOWWAIT. The application is processing data for the database manager. Tells the user that application code is in the process of running.

WAITFOR REMOTE. A partitioned database instance is standing by for a reply from a remote partition.

Average Active Agents

Data displayed	**agents/s.** The number of active agents per second (Average active sessions).

Active/Inactive Agents

Data displayed	**Active.** The number of active agents.
	Inactive. The application is at one of three statuses: Connected, Wait or Decoupled.

Foreground/Background Agents

Data displayed	**Background.** Sum of background application agents.
	Foreground. Sum of foreground application agents

DB2 CPU Usage

Data displayed	**DB2 CPU Usage.** Total CPU usage by all processes running on the monitored database members.
	Clicking the metric displays a pop-up graph.

Agents

Data displayed	**Count.** Displays the number of running agents during the specified time range.

Current Agent List

Displays detailed agent metrics.

| | Data displayed | | • **Kill Session.** When the agent was stopped. |

Data displayed

- **Kill Session.** When the agent was stopped.

- **AGENT ID.** ID of the agent.

- **Member Number.** Member number of the agent.

- **Application ID.** Application ID of the agent.

- **Logon Time.** Time the agent logged in.

- **Application Status.** Agent status.

- **Stmt Text.** Text describing the agent status.

- **Active Time.** Time the agent became active.

- **Application Name.** The name of the application running at the client.

- **CPU.** The total CPU time used by database manager agent process.

- **Client PID.** The process ID of the client application that made the connection to the database.

- **Coord PID.** The process ID (UNIX systems) or thread ID (Windows systems) of the coordinator agent for the application.

- **Assoc Agents.** The number of subagents for all applications.

User-defined Collections

Displays user-defined metrics tailored to your database monitoring requirements. For procedures on defining user-defined collections, see Adding User-defined Collections.

How to Get Here

On the **Database home page** toolbar, click **Activity** and select **User-defined Collections**.

Description of the Pane

Data displayed

Collections. Name of the collection created. For procedures on defining User-defined Collectors, see Adding User-defined Collections.

- **Query's Full SQL Text.** The SQL text used to define the user-defined collection.

- **Collection response time.** The time it took to collect to respond in milliseconds.

Description of the Pane

- **Display value for.** Allows you to view the SQL query's metrics, either for the last sample (Last Sample Taken) or for the entire specified time range (Selected Period).
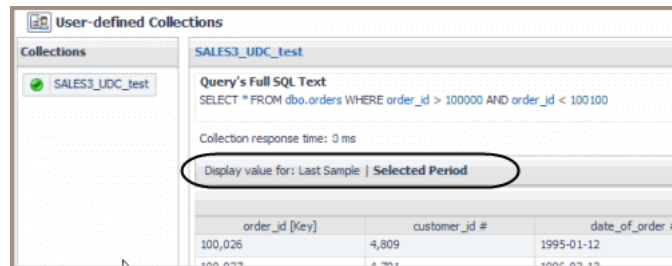
The Selected Period is based the time range selected in the Foglight Timeline.
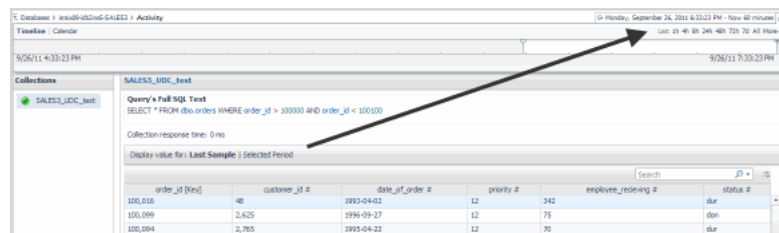
Description of the Pane

- • **SQL Metrics Table.** This table's columns display the metrics of the SQL collection defined and running.
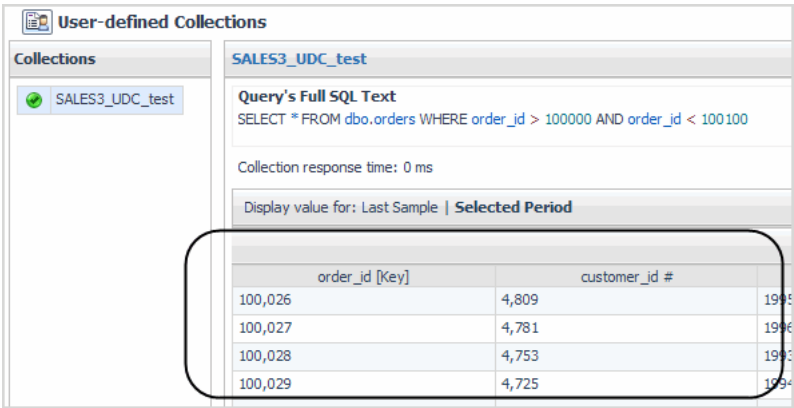
Clicking a metric in a column displays a pop-up of that metric with a graph of the average value plotted over time and raw data samples.
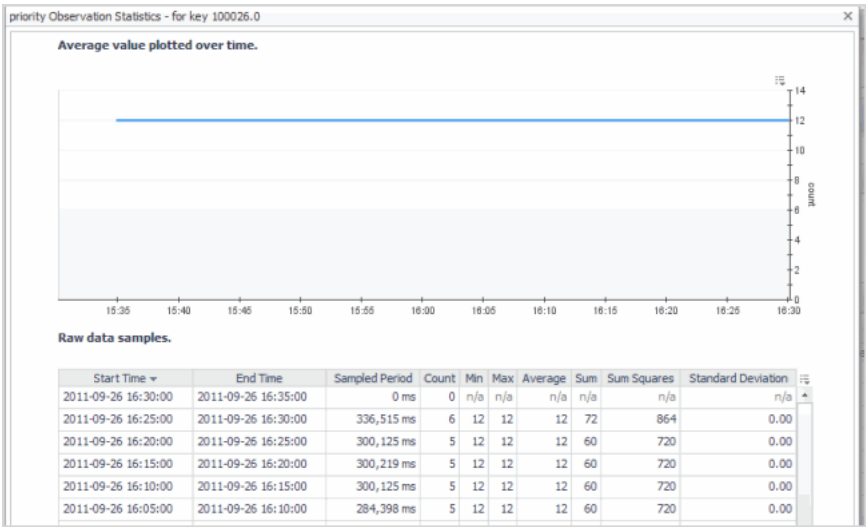
Where to go next

Memory Drill-downs

The Memory drill-down provides access to:

- • Memory Summary Pane
- • Buffer Pools Pane

Memory Summary Pane

Provides memory summary metrics on the members and memory pools for the selected database.

Select Whole Database or a member within the selected database from the list at the top of the pane.

How to Get Here

On the **Database home page** toolbar, click **Memory** and select **Memory Summary**.
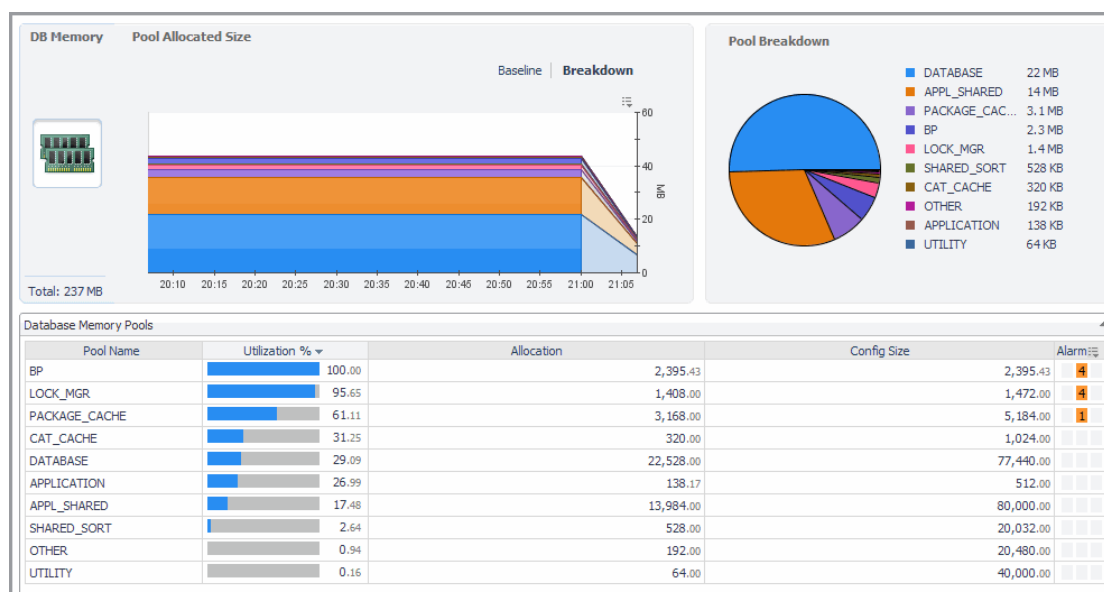
Description of the Pane

Data displayed — Charts	• **DB Memory.** Displays the total memory allocated to the selected database.
	• **Pool Allocated Size.** Displays a breakdown of the total memory allocated to the members of the selected database using a bar chart.
	• **Pool Breakdown.** Displays a breakdown of the total memory allocated to the members of the selected database using a pie chart.
Data Displayed — Database Member Memory Pools List	• **Pool Name.** The name of the memory pool.
	• **Utilization %.** The percentage of configured allocated memory that is in use.
	• **Allocation.** The average allocated size (in KB) of a memory pool.
	• **Config Size.** The internally configured size (in KB) of a memory pool in the DB2 database system.
	• **Alarm.** The alarm count and state of the member or memory pool. Click an alarm count to view information about the alarm on the Outstanding Alarms dialog box. For complete information about alarms in Foglight, see the *Foglight User Help*.

Buffer Pools Pane

Provides performance metrics on the buffer pools for the selected database.

How to Get Here

On the **Database home page** toolbar, click **Memory** and select **Buffer Pools**.

This view provides the following embedded views:

- Buffer Pools — Details tab
- Buffer Pools — Activity tab
- Buffer Pools — Direct Activity tab

Buffer Pools — Details tab

The Details tab displays a top-level summary of the buffer pools. The tabs in the lower portion of the view display graphs of the various metrics over time.

Description of the Pane

Data displayed	
	• **Buffer Name.** The name of the monitored buffer pool.
	• **Total Hit Ratio.** A measure of how often a page access (getpage) is satisfied without requiring an I/O operation.
	• **Tablespaces.** The number of tablespaces using this buffer pool.
	• **Data Hit Ratio.** A measure of how often a data page access (getpage) is satisfied without requiring an I/O operation.
	• **Index Hit Ratio.** A measure of how often an index page access (getpage) is satisfied without requiring an I/O operation.
	• **XDA Hit Ratio.** A measure of how often an XML storage object (XDA) page access (getpage) is satisfied without requiring an I/O operation.
	• *pureScale only* — **GBP Hit Ratio**. A measure of how many times a requested page was found in the GBP out of all times it was requested.
	• **Col Hit Ratio**. A measure of how often a column-organized page access is satisfied without requiring an I/O operation. Available for version 10.5 and above.

Description of the Pane

- **Read Time.** Indicates the total amount of time (in seconds) spent reading in data and index pages from the tablespace containers (physical) for all types of tablespaces.

- **Write Time.** Provides the total amount of time spent (in seconds) physically writing data or index pages from the buffer pool to disk.

- **Direct Reads Rate.** The number of read operations that do not use the buffer pool, per second.

- **Direct Writes Rate.** The number of write operations that do not use the buffer pool, per second.

- **Direct Read Time.** The elapsed time (in seconds) required to perform the direct reads.

- **Direct Write Time.** The elapsed time (in seconds) required to perform the direct writes.

- **Files Closed.** The total number of database files closed. This element can be used to help with determining the best value for the `maxfilop` configuration parameter.

- **Unread Prefetch Pages.** Indicates the number of pages that the prefetcher read in that were never used. If this number is high, prefetchers are causing unnecessary I/O by reading pages into the buffer pool that are not being used.

- **Alarms.** The alarm count and state of the member or memory pool. Click an alarm count to view information about the alarm on the Outstanding Alarms dialog box. For complete information about alarms in Foglight, see the *Foglight User Help*.

- **Size**. The current buffer pool size in pages.

- **Col Logical Reads**. Indicates the number of column-organized pages requested from the buffer pool (logical) for regular and large table spaces.

  Available for version 10.5 and above.

- **Col Logical Reads Rate**. Indicates the number of column-organized pages which have been requested from the buffer pool (logical) for regular and large table spaces per second.

  Available for version 10.5 and above.

- **Col Physical Reads**. Indicates the number of column-organized pages read in from the table space containers (physical) for regular and large table spaces.

  Available for version 10.5 and above.

- **Col Physical Reads Rate** - Indicates the number of column-organized pages read in from the table space containers (physical) for regular and large table spaces, per second .

  Available for version 10.5 and above.

- **Automatic**. Indicates whether the buffer pool has self-tuning enabled. 1 indicates self-tuning is enabled for the buffer pool, and 0 otherwise.

Buffer Pools — Activity tab

Displays graphs of read and write metrics for the buffer pools against a baseline.

Description of the Pane

Data Displayed	
	- For **Read Time** and **Write Time** descriptions, see Buffer Pools — Details tab.
	- **Total Logical Reads Rate.** The total number of logical reads (index, XDA, and data) per second in the buffer pool.

Description of the Pane

- **Total Physical Reads Rate.** The total number of physical reads (index, XDA, and data) per second in the buffer pool.

- **Data Logical Reads Rate.** The number of data logical reads per second, which includes accesses to data pages that are already in the buffer pool when the database manager needs to process the page. Also includes data pages that are read into the buffer pool before the database manager can process the page.

- **Data Physical Reads Rate.** The number of data pages read in per second from the tablespace containers (physical) for regular and large tablespaces.

- **Index Logical Reads Rate.** The number of index logical reads per second, which includes accesses to index pages that are already in the buffer pool when the database manager needs to process the page. Also includes index pages that are read into the buffer pool before the database manager can process the page.

- **Index Physical Reads Rate.** The number of index pages read in per second from the tablespace containers (physical) for regular and large tablespaces.

Buffer Pools — Direct Activity tab

Displays graphs of direct reads and direct writes metrics to the buffer pools. For metric descriptions, see Buffer Pools — Details tab.

Storage Drill-downs

The Storage drill-down provides access to:

- Tablespaces

- Tablespaces Buffer Pool Activities

Tablespaces

The Tablespaces Pane provides metrics and graphs of the top 10 tablespace activities and summarizes tablespace usage. This pane provides the following views:

- Top 10 Tablespaces

- Tablespaces Summary
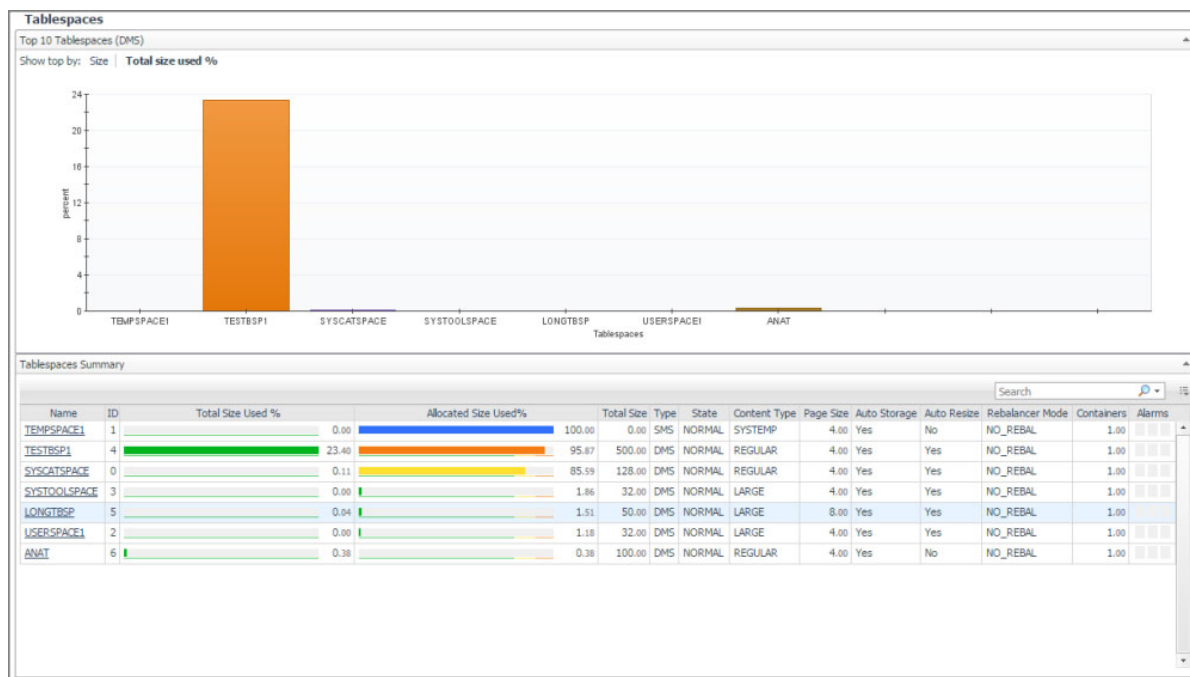
How to Get Here

On the **Database home page** toolbar, click **Storage** and select **Tablespaces**.

You can also access the Tablespaces pane for a database from the Databases Dashboard. Click the icon at the far right of the database name cell and selecting **Tablespaces** from the menu that appears.

Top 10 Tablespaces

Displays the 10 tablespaces which have the highest size or the highest percentage of space utilization, according to the user's selection. The default selection for sorting graphically is space utilization. Selecting the alternate value updates the chart and the table.
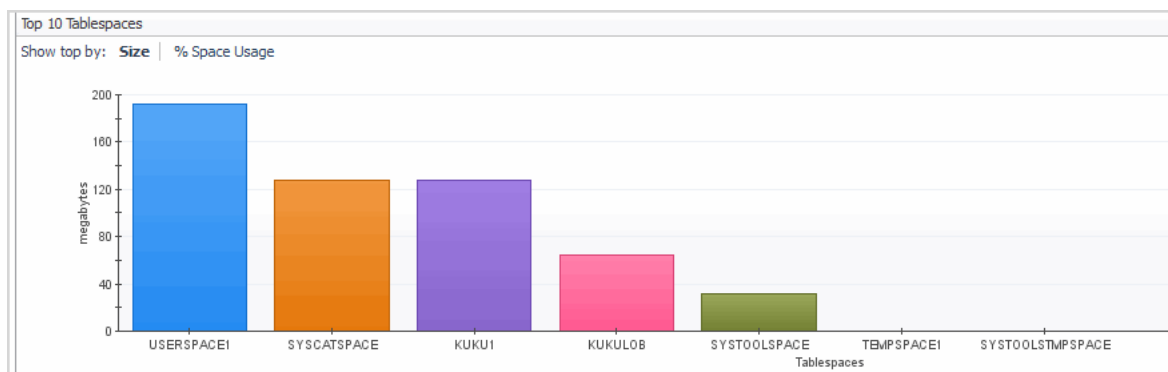
Description of the View

Data displayed	• **Show top by.** Use the **Show top by** options to select metrics which display graphs for size and percentage of tablespace usage. Click the required metric representation to display the respective graph. The following graphs are provided: • **Size**: Tablespace size over the last 30 days or during the timeframe selected (which ever is longer). • **% Space Usage**. Percent of the tablespace in use. Utilization is calculated after taking into account the type of the tablespace (SMS/DMS), its parameters (like auto extend), the free space on the file system and the IBM architecture size limitations. •
Axis Definitions:	• **Percent (chart vertical axis).** The percentage of the database tablespace pages currently being utilized. • **Megabytes (chart vertical axis).** The database tablespace utilized size in megabytes. • **Tablespaces (chart horizontal axis).** The names of the top 10 most utilized tablespaces. • Clicking a specific tablespace representation drills down to the Tablespace-specific page.
Where to go next	 • For more information, see Tablespace-specific Page on page 81.
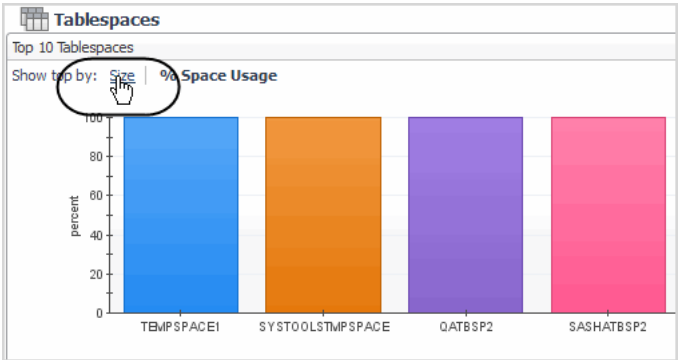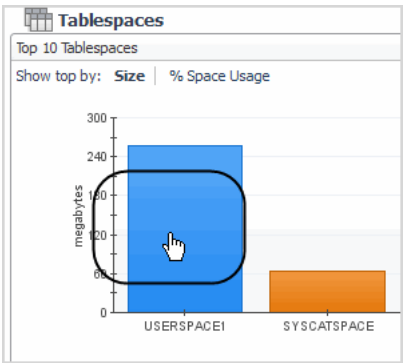
Tablespaces Summary

The Tablespaces Summary view shows the status for all the database tablespaces, including space usage in percent and state of the tablespace.

How to Get Here

The Tablespace Summary view is located in the lower part of the Tablespaces pane.
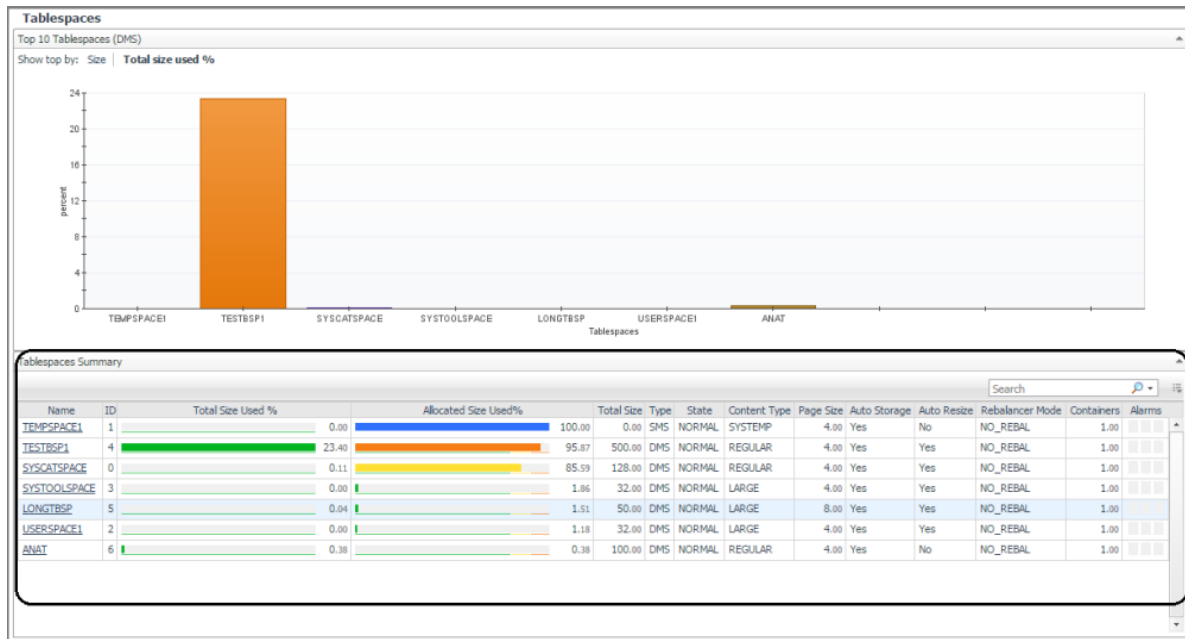
Description of the View

Data displayed	
	• **Allocated Size used%".** The percentage of tablespace pages used out of all usable allocated pages.
	• **Total Size Used %.** Tablespace utilization % including free space on the file sytstem and DB2 architecture limits
	• **Alarms**. The number of warning, critical, and fatal alarms for this DB2 database tablespace. The dwell shows the most recent alarms raised against this tablespace, sorted by severity. Click this field to view the Alarms list. This list allows browsing through the alarms, which are sorted according to severity. See the *Foglight Online Help, Monitoring System-Wide Alarms* for details on the alarm information.
	• **Auto Resize.** Relates to the DMS tablespaces and storage tables stored automatically. A setting of one indicates enablement of the auto resizing.
	• **Auto Storage.** Describes whether the tablespace was created as an automatic storage tablespace. A value of 1 (Auto) means yes; 0 means no.
	• **Containers.** Container count.
	• **Content Type.** Text identifier defining the size or type of the tablespace: ANY, LARGE, SYSTEMP, or USRTEMP.
	• **ID.** Numerical identification of the tablespace.
	• **Name.** The name of the tablespace. Clicking the tablespace name drills down further to the Tablespace-specific page.
	• **Page Size.** Total page size in KBs.
	• **Rebalancer Mode.** Indicates if the current re-balancing process is removing space from a tablespace or adding space to a tablespace.
	NOTE: Only applicable to a DMS tablespace.
	• **State.** Text identifier of the tablespace state.
	• **Total Size.** The total amount of allocated space (in megabytes) in this tablespace.
	• **Type.** Text identifier of DMS or SMS.

Description of the View

- **MAX ARCHITECTURE SIZE TB** - Maximum size allowed by DB2 for a tablespace as reported in the DB2 docs. Size is given in TB. -1 indicates limit is not relevant

- **MAX ARCHITECTURE SIZE pages** - Maximum size allowed by DB2 for a tablespace as reported in the DB2 docs . Size is given in Mega pages (1 means 1 million pages). -1 indicates limit is not relevant.

- **FS Free Size MB** - Total amount of free space left on all file system used by the tablespace. Size is given in MB

Clicking the tablespace name drills down to the Tablespace-specific page.

Where to go next

For more information, see

Tablespace-specific Page

This page provides details on tablespace properties and space utilization, allowing you to analyze any issues affecting the selected member.
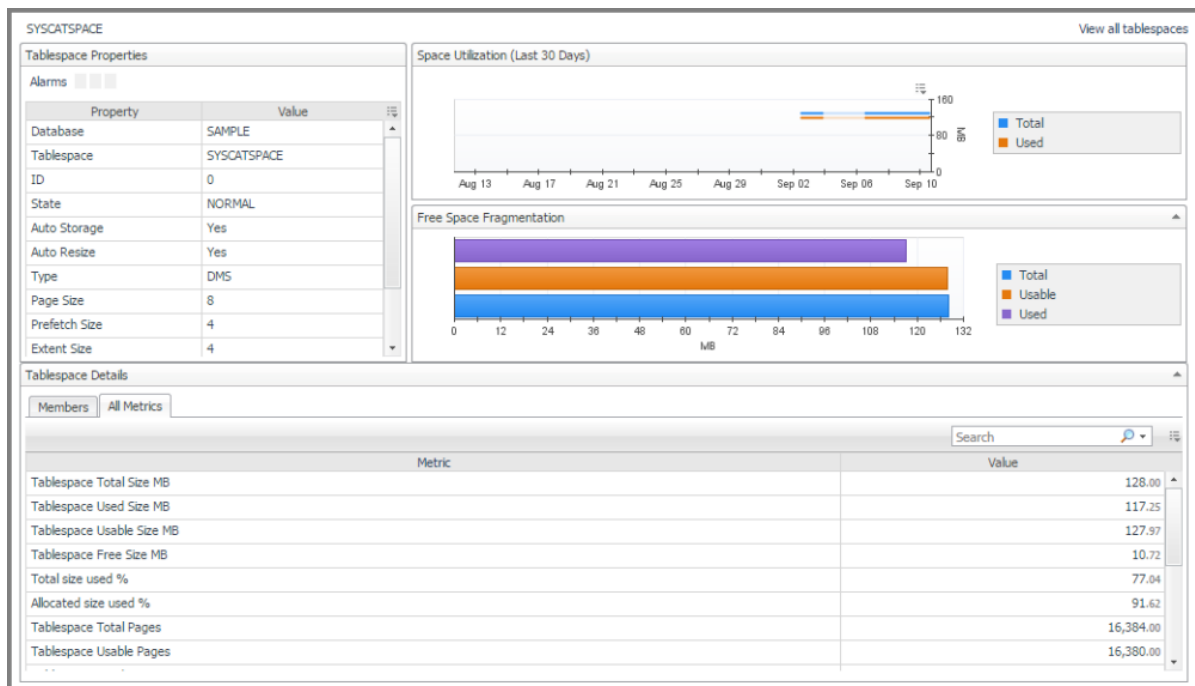
How to Get Here

From the **Tablespaces** pane, click the required tablespace in the **Top 10 Tablespaces** view or the tablespace name in the **Tablespace Summary** view.
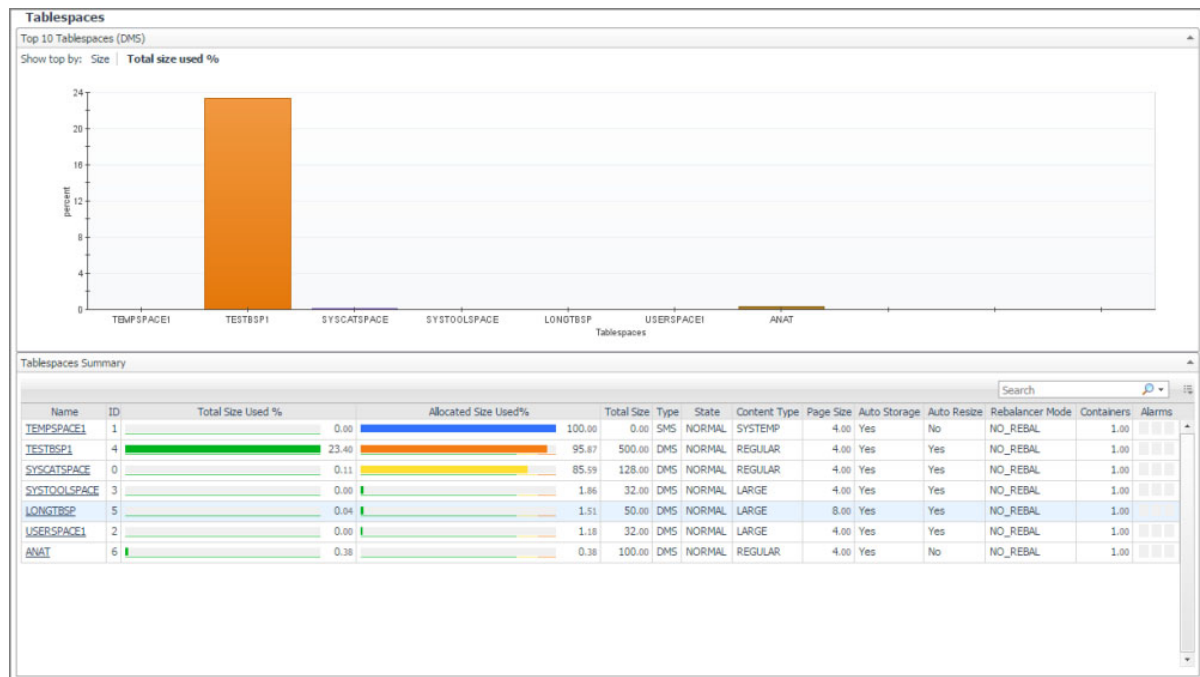
Description of Views

This pane is made up of the following views.

- Tablespace Properties
- Space Utilization
- Free Space Fragmentation
- Tablespace Details
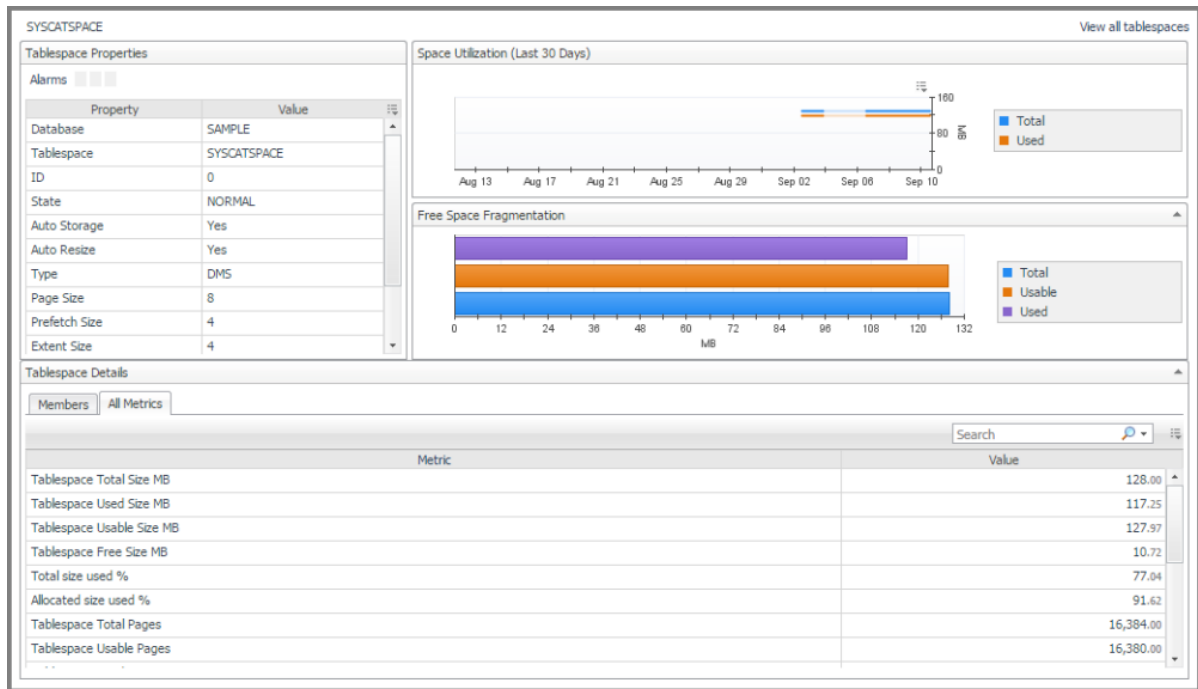
Tablespace Properties

This view provides the following panes:

- Tablespace properties alarms

- Tablespace properties table

	• **Alarms.** The number of warning, critical, and fatal alarms for this database tablespace. The dwell shows the most recent alarms raised against this tablespace, sorted by severity.
Data Displayed	
	Clicking an alarm displays an alarm management pop-up.
	• **Auto Resize.** Relates to the DMS tablespaces and storage tables stored automatically. A setting of 1 indicates enablement of the auto resizing.
	• **Auto Storage.** Reports if the page is automatically stored.
	• **Content Type.** Text identifier defining the size or type of the tablespace: ANY, LARGE, SYSTEMP, or USRTEMP.
	• **Database.** Database name.
	• **Extent Size.** Size of the tablespace extent.

- **Page Size.** Total page size in KBs.

- **Prefetch Size.** The highest number of pages a pre-fetcher receives from a disk for a given time. If this function is enabled, a value of -1 displays.

- **Rebalancer Mode.** Shows the direction of rebalancing and whether this process is adding or removing tablespaces.

- **State.** Text identifier of the database state.

- **Tablespace.** Tablespace name.

- **Type.** Text identifier of DMS or SMS.

Space Utilization

Description	Provides a chart of the total utilized space plotted over time.
	Total. Total size allocated for the tablespace in MB.
	Used. The amount of MB that are in use for a table space. For an SMS table space it is equal to tablespace total size. Size is given in MB.

Free Space Fragmentation

Description	Displays a status of the free fragmentation space.
	• **MB.** Measurement of the graph in megabytes.
	• **Total.** Total free space.
	• **Usable.** Usable free space.
	• **Used.** Space used.

Tablespace Details

The Tablespace Details view provides the following tabs:
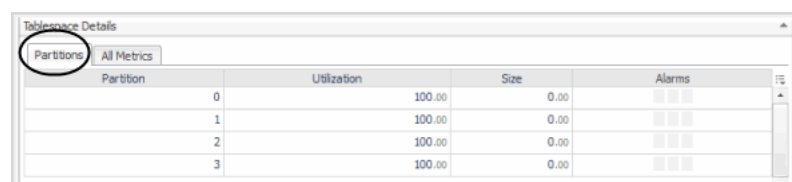
- Members
- All Metrics

Members

Description of the Members Tab

Provides metrics on member utilization, size, and alarms affecting the member.
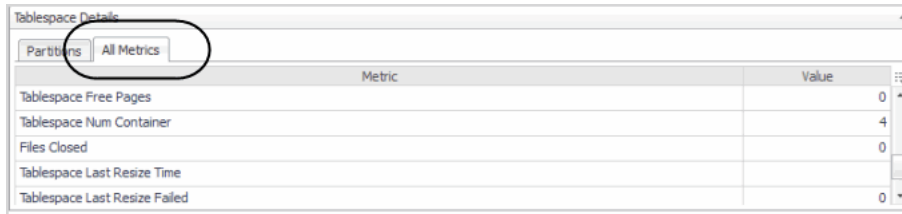
Description	
	• **Alarms.** Any alarms that fired and are affecting the tablespace. Clicking an alarm displays the alarms management pop-up.
	• **Member.** The database member from which the data was retrieved.

Description of the Members Tab

- **Size.** Total size of the tablespace in MBs.

- **Utilization.** Overall utilization in percent. Auto storage tablespace utilization is calculated by the used space compared to the allocated tablespace size, not the total mount point size.

All Metrics

Description of the All Metrics Tab

Name	Description
Description	The All Metrics tab provides tablespace properties, space utilization, and fragmentation metrics.

- **Files Closed.** Files that have been closed in the tablespace.

- **Tablespace Free Pages.** Total number of free pages in the tablespace.

- **Tablespace Free Size MB.** Amount of free space.

- **Tablespace Last Resize Failed.** The time the tablespace failed during its latest resize attempt.

- **Tablespace Last Resize Time.** When the tablespace was last resized.

- **Tablespace Num Container**. Number of containers available in the tablespace.

- **Tablespace Total Pages.** Total number of pages in the tablespace.

- **Tablespace Total Size MB.** Total tablespace size.

- **Tablespace Usable Pages.** Total number of usable pages in the tablespace.

- **Tablespace Usable Size MB.** Total amount of space available.

- **Tablespace Used Pages.** Total number of pages used by the tablespace.

- **Tablespace Used Size MB.** Total amount of space used.

- **Allocated Size used%.** The percentage of tablespace pages used out of all usable allocated pages.

- **Value**. Value of the metric reported.

- **Total Size Used %.** Tablespace utilization % including free space on the file sytstem and DB2 architecture limits

- **MAX ARCHITECTURE SIZE TB** - Maximum size allowed by DB2 for a tablespace as reported in the DB2 docs. Size is given in TB.

- -1 indicates limit is not relevant

- **MAX ARCHITECTURE SIZE pages** - Maximum size allowed by DB2 for a tablespace as reported in the DB2 docs . Size is given in Mega pages (1 means 1 million pages).

- -1 indicates limit is not relevant.

- **FS Free Size MB** - Total amount of free space left on all file system used by the tablespace. Size is given in MB

Tablespaces Buffer Pool Activities

Provides the metrics required to monitor tablespace buffer pool activities and can be sorted according to the buffer pool hit ratio.
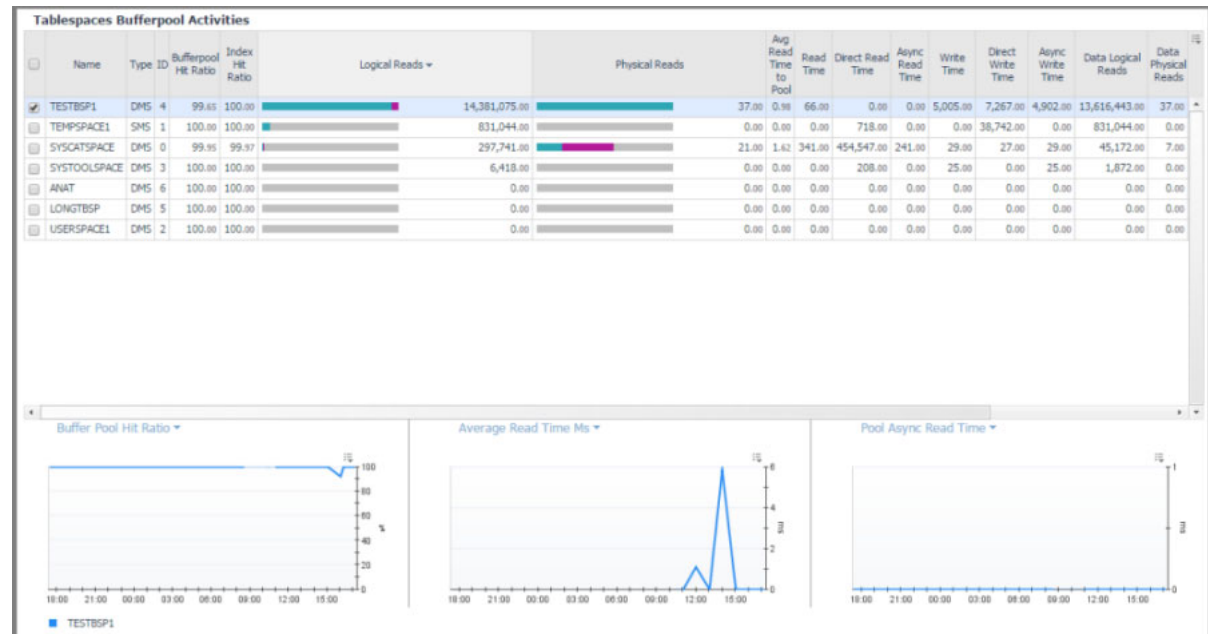
How to Get Here

On the **Database home page** toolbar, click **Storage** and select **Tablespaces Bufferpool Activities**.

This pane provides the Tablespaces Bufferpool Activities view.

Tablespaces Bufferpool Activities

Displays the top tablespaces buffer pool activities metrics and allows you to choose the tablespace to plot and view charts on the details and performance of the selected tablespace. The tablespaces are sorted according to the buffer pool hit ratio.

Description of the View

Data displayed	
	• **Async Read Time.** The time (in milliseconds) spent on reading data pages by database manager prefetchers.
	• **Async Write Time.** The time (in milliseconds) spent on writing data or index pages by database manager prefetchers.
	• **Bufferpool Hit Ratio.** The percentage of read requests that are satisfied by the buffer pool without having to physically read from the disk.
	• **Data Logical Reads.** The number of logical read requests for data pages that go through the buffer pool.
	• **Data Physical Reads.** The number of physical (disk) read requests for data pages that do not go through the buffer pool.

Description of the View

- **Data Writes.** The number of data writes.
- **Direct Read Time.** The time (in milliseconds) spent on performing direct reads.
- **Direct Write Time.** The time (in milliseconds) spent on performing direct writes.
- **Index Hit Ratio.** The percentage of read requests for index pages that are satisfied without having to read from the disk.
- **Index Logical Reads.** The number of logical read requests for index pages that are satisfied without having to read from disk.
- **Index Physical Reads.** The number of physical (disk) read requests for index pages that need to read data from disk.
- **Index Writes.** The number of index writes.
- **Name.** The tablespace name.
- **Read Time.** The amount of elapsed time (in milliseconds) spent on processing read requests.
- **Type.** The tablespace type (DMS Database-Managed Space) or SMS (System-Managed Space).
- **Write Time.** The amount of elapsed time (in milliseconds) spent on processing write requests.
- **Total logical reads.** Total pages read from the bufferpool. count all page types : data, index, XDA and column-organized .
- **Total physical reads.** Total pages read from the table space containers (physical) to the pool. Count all page types : data, index, XDA and column-organized.
- **Col logical Reads.** Total number of column-organized pages requested from the buffer pool (logical).
- Relevant from version 10.5 and above.
- **Col Physical Reads.** Total number of column-organized pages read in from the table space containers (physical).
- Relevant from version 10.5 and above.
- **Col Hit Ratio.** Percentage of column-organized pages reads that were carried out from the buffer pool for the selected tablespace, without having to perform physical reads.
- calculation is -
  100 * (1 - (col_physical_reads / col_logical_reads)))
- Relevant from version 10.5 and above
- **GBP Logical Reads.** The number of pages read from the Group Buffer Pool (GBP) because the page was either invalid or not present in the Local Buffer Pool (LBP). Only relevant for a puresclae instance.
- **GBP physical reads.** The number of Group Buffer Pool (GBP) dependent pages that were read into the local buffer pool from disk because it was not found in the GBP.
- Only relevant for a puresclae instance.
- **GBP Hit Ratio.** The percent of time a requested page was found in the group bufferpool out of all times it was requested. If a page is found in the GBP it saves the database the time to look for it in the Local Buffer Pool. Only relevant for a puresclae instance.

Description of the View

- **Avg Read Time to Pool.** The average time it takes to read a single page from the table space containers (physical) to the buffer pool. value is given in milliseconds.

Where to go next	One or more tablespace entries can be viewed by selecting them in the table. The graphs in the lower pane sho the information for the selected tablespaces.

Log Drill-down

The Log drill-down provides a diagnostic graph and log of database log activity. Records displayed are those configured in the Diagnostic Log administration screen.
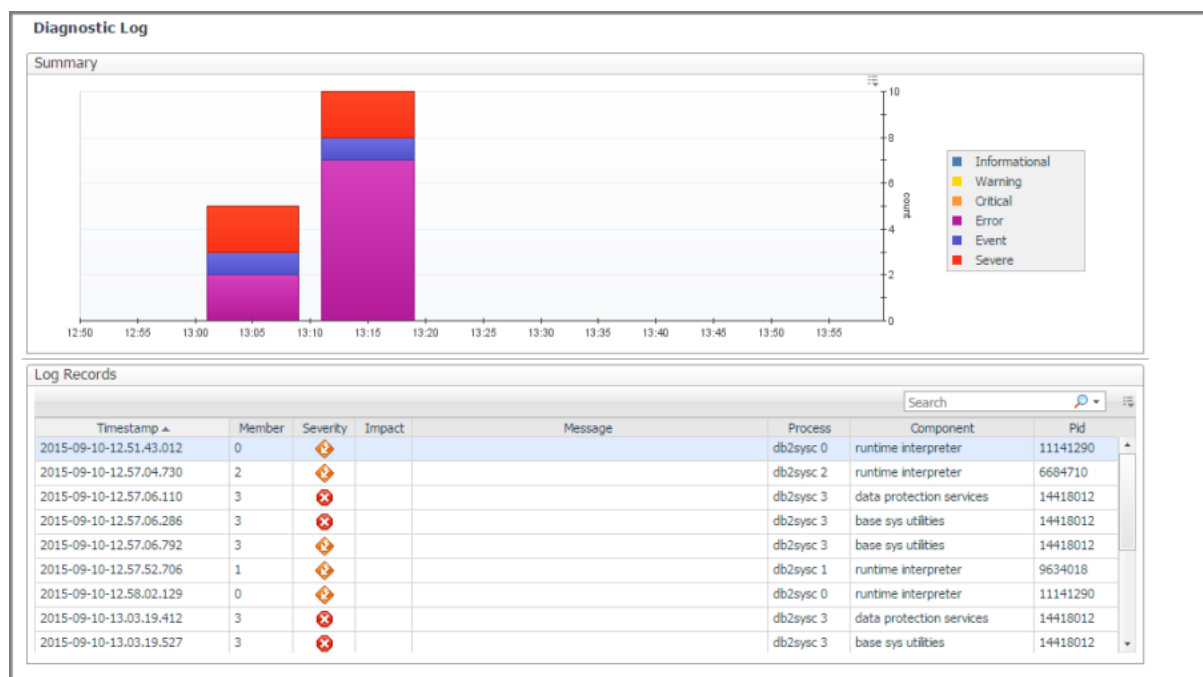
How to Get Here

On the **Database home page** toolbar, click **Log**.

This pane provides the following views:

- Summary
- Log Records

Summary

Description	Displays a graphical history of the informational, warning, critical, error and other counts graphed over a period.

Log Records

Description	Displays information log table messages in tabular form.

- **Timestamp.** The exact time and date when the message log was created.
- **Member.** Member where the message came from.
- **MSGNUM.** Message number
- **Severity.** Severity of the message.
- **Type.** Message type.
- **Message.** Text of the message.
- **Pid.** Process ID.
- **Process.** Which DB2 process triggered the message.
- **TID.** Timestamp identification number.
- **Application ID.** Identifies the application from which the message originated.

Selecting any record will open a pop-up with the full log record.

Configuration Drill-down

The Configuration drill-down provides the list of initialization parameters with their current values and other pertinent information, sorted in alphabetical order.

How to Get Here

On the **Database home page** toolbar, click **Configuration**.

This pane provides the Initialization Parameters view.

Initialization Parameters

Description	Displays the list of initialization parameters with their current values and other pertinent information, sorted in alphabetical order.
	• **Parameter.** The initialization parameter name.
	• **Current Value.** The current value of the initialization parameter.
	• **Flags.** Whether the initialization parameter is flagged as AUTOMATIC.
	• **Deferred Value.** The deferred value of the initialization parameter. If a non-AUTOMATIC parameter was changed and the database has not yet been recycled, the deferred value may be different than the current value.
	• **Type.** The type of value for the initialization parameter.
	• **Modified.** Whether the initialization parameter has been modified.
Where to go next	Clicking any table cell displays the change history for the corresponding initialization parameter.

Member Overview home page

This home page provides metrics on the state of the member over the course of the selected time range. Pop-ups provide graphs that show you how the member is performing and whether there are any critical areas that need to be addressed.

The metrics provided on this home page are different for DB2 v9.7.0.1 and later than they are for DB2 databases earlier than v9.7.0.1. For more information, see below sections.

How to Get Here

1 From the Database home page, click **Whole Database** and select the member you want.

2 Click **Summary** and select **Overview**.

From the Member (Real Time) home page, you can drill down to view additional metrics for the selected DB2 member or partition.

The following summary and drill-downs pages are provided:

* Summary

* SQL Performance (links to description under Database home page)

* Activity Drill-downs (links to description under Database home page)

* Memory Drill-downs (links to description under Database home page)

* Storage Drill-downs (links to description under Database home page)

* Log Drill-down (links to description under Database home page)

* Configuration Drill-down (links to description under Database home page)

Summary

This home page contains the following panes:

* Workload Information

* Storage Information

* OS Performance Information

* Availability Information

- Cache Hit Ratio Information
- HADR Information

Workload Information

The Workload pane displays the session activity and CPU usage information for the selected partition. It is at the upper left of the Partition (History) home page.

Description of the Pane

Data displayed for DB2 databases v9.7.0.1 and later	• **Request Time Breakdown:**
	
	  ▪ Baseline — A comparison of the average request time with the expected (according to the baseline) request time, plotted over the course of the selected time range.
	
	  ▪ Breakdown — the average request time by event category, plotted over the course of the selected time range.

	• **CPU Wait Time.** The average CPU wait time (in milliseconds), plotted over the course of the selected time range.
Data displayed for DB2 databases earlier than v9.7.0.1	• **Average Active Sessions.** A comparison of the average number of active sessions over the course of the selected time range with the average number of active sessions expected (according to the baseline).
	• **CPU Usage.** The percentage of CPU used by the partition, plotted over the course of the selected time range.
Where to go next	Mousing over a data point on either of the charts displays a pop-up that provides more detailed information about the data point. Clicking the **Investigate using the "Top SQL" drilldown** link takes you to the Top SQL pane for additional information.

Storage Information

The Storage pane displays the storage information for the selected partition. It is at the upper right of the Partition (History) home page.

Description of the Pane

Data displayed	• **Storage Graph.** The storage activity of the partition plotted over the course of the selected time range.
	
	The storage can be plotted as either:
	
	  • Incremental change growth — incremental changes in partition growth (in MB/s) plotted over the course of the selected time range
	
	  • Database size — the size of the database (in GB) plotted over the course of the selected time range

	• **Avg. Consumption (MB/hour).** The average storage space consumed (in MB/hour) over the course of the selected time range.
	• **Total Allocated (MB).** The total storage space allocated (in MB).
Where to go next	Mousing over a data point on the plot displays a pop-up that provides more detailed information about the data point. Clicking the **Investigate using the "Table Space" drilldown** link takes you to the Tablespaces pane for additional information.

OS Performance Information

The OS Performance pane displays the OS performance information for the selected partition. It is at the bottom left of the Partition (History) home page.

Description of the Pane

Data displayed	• **Avg. CPU (%).** The average CPU consumed, as a percentage of the total allocated, over the course of the selected time range.
	• **Avg. Memory (%).** The average memory consumed, as a percentage of the total allocated, over the course of the selected time range.
	• **Disk Utilization (%).** The disk utilization, as a percentage of the total allocated, over the course of the selected time range.
Where to go next	Clicking any of the three charts displays a pop-up that provides additional, related information. On the pop-up, clicking the **Investigate using the "Host Monitor" drilldown** link takes you to the Host Monitor dashboard for additional information.

Availability Information

The Availability pane displays the availability information for the selected partition. It is at the lower middle of the Partition (History) home page.

Description of the Pane

Data displayed	• **Sessions.** The availability of the active and inactive sessions plotted over the course of the selected time range.
	• **Member Availability.** The average availability of the partition over the course of the selected time range.
For DB2 databases v9.7.0.1 and later	• **Average Request Time (ms).** The average length of time (in ms) it took to process a single request, over the course of the selected time range.
For DB2 databases earlier than v9.7.0.1	• **Transaction Rate.** The number of transactions being processed per second.
Where to go next	Mousing over a data point on the Sessions chart displays a pop-up that provides more detailed information about the data point. Clicking anywhere on the Partition Availability chart displays a pop-up that provides additional, related information. Clicking the icon next to the Avg. Request Time or Transaction Rate metric displays a pop-up that provides additional, related information.

Cache Hit Ratio Information

The Cache Hit Ratio pane displays the cache hit ratio information for the selected partition. It is at the right of the Partition (History) home page.

Description of the Pane

Data displayed	• **Total BP (%).** The percentage of database page I/O requests that were satisfied using the buffer cache (and therefore the database did not have to perform disk reads), over the course of the selected time range.
	• **Package (%).** The number of times that a requested section was not available for use and had to be loaded into the package cache, as a percentage of the total number of times that a section was requested, over the course of the selected time range.

Description of the Pane

	• **Catalog (%).** The number of times that a table descriptor or authorization information was not found in the catalog cache and had to be inserted there, as a percentage of the total number of times that a table descriptor or authorization information was requested, over the course of the selected time range.
Where to go next	Clicking the icon next to any of the cache hit ratios displays a pop-up that provides additional, related information. Clicking the link at the bottom of any of the three pop-ups takes you to the corresponding dashboard for additional information.

HADR Information

If the database is configured for HADR, the HADR pane displays the associated HADR metrics for the partition. It is at the bottom right of the Partition (History) home page. For more information on these parameters, see **HADR Activity Collection** on page 128.

Description of the Pane

Data displayed	• **Role.** The HADR role of the database.
	• **HADR Remote Host.** The name of the HADR remote host.
	• **HADR Remote Instance.** The name of the HADR remote instance.
	• **Log Gap.** The size (in KB) of the gap between the primary log sequence number and the standby log sequence number.

	• **HADR State.** The HADR state of the database:
	  ▪ Disconnected
	  ▪ Performing local catch-up
	  ▪ Waiting to connect to its partner to perform remote catch-up
	  ▪ Performing remote catch-up
	  ▪ In a peer state

Where to go next	Clicking the icon next to a metric displays a pop-up with additional, related information.

Member (Real Time) home page

This home page provides metrics on the current state of the selected partition. It allows you to view how the partition is performing and any performance or bottleneck issues affecting the partition. Pop-ups provide additional information and graphs.
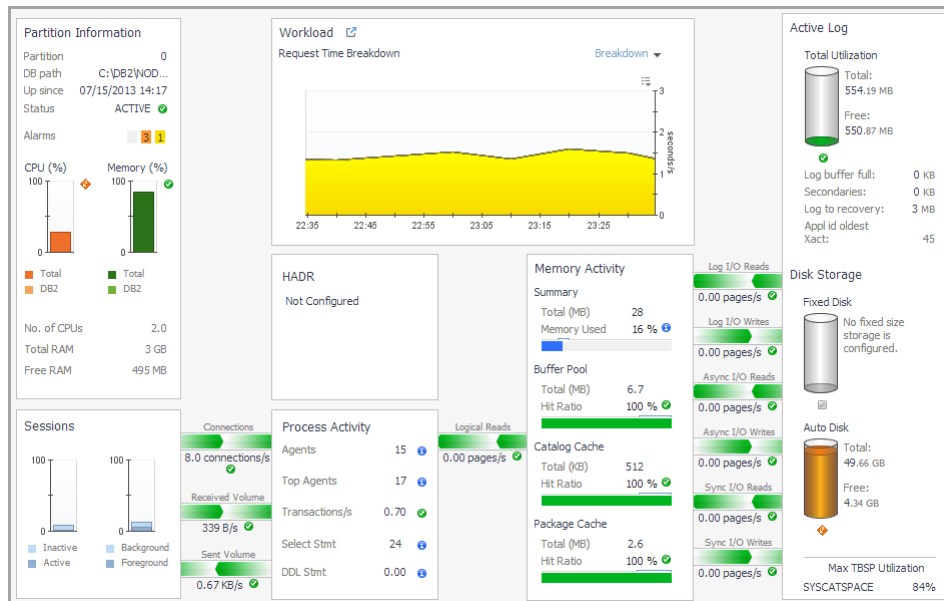
How to Get Here

1   From the Database home page, click **Whole Database** and select the partition you want.
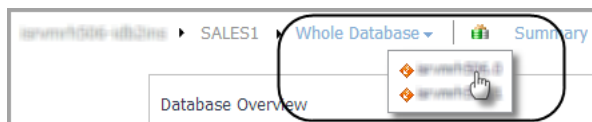
2   Click **Summary** and select **Real Time**.

From the Member (Real Time) home page, you can drill down to view additional metrics for the selected DB2 member or partition.

The following summary and drill-downs pages are provided:

- Summary

- SQL Performance (links to description under Database home page)

- Activity Drill-downs (links to description under Database home page)

- Memory Drill-downs (links to description under Database home page)

- Storage Drill-downs (links to description under Database home page)

- Log Drill-down (links to description under Database home page)

- Configuration Drill-down (links to description under Database home page)

Summary

This home page contains the following panes:

- Member Information
- Sessions
- Workload Information
- HADR
- Process Activity
- Memory Activity
- I/O Activity
- Active Log
- Disk Storage

Member Information

The Member Information pane displays member status and connection information, member CPU and memory information, and member alarms. It is at the upper left of the Member (Real Time) home page.

Description of the Pane

Data displayed	
	• **Member.** The identifying number for the partition.
	• **DB path.** The path of the partition.
	• **Up since.** The date and time the partition connected.
	• **Status.** The status of the partition.
	• **Alarms.** The alarms (due to partition irregularities) that are currently outstanding against the partition.
	• **CPU (%).** The total CPU consumption by processes running on the host (including any DB2 processes), and the total CPU consumption by the DB2 processes running on the host.
	• **Memory (%).** Total memory consumption by all processes running on the host (including DB2 processes), and the total memory consumption by the DB2 processes running on the host.
	• **No. of CPUs.** The number of CPUs allocated to the partition.
	• **Total RAM.** The total RAM allocated to the partition.
	• **Free RAM.** The total free RAM available for the partition.
Where to go next	Clicking the icon next to the Status displays a pop-up that provides additional, related information.
	Mousing over a data point on either of the CPU or Memory charts displays a pop-up that provides additional information about the data point.
	Clicking the CPU or Memory charts displays a pop-up that provides additional, related information.
	Mousing over or clicking an alarm count displays a pop-up that provides additional, alarm-related information.

Sessions

The Sessions pane displays the session distribution information for the selected partition. It is at the lower left of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Inactive or Active.** The distribution of currently running inactive and active sessions.
	• **Background or Foreground.** The distribution of currently running background and foreground sessions.
	• **Connections.** The current number of active connections.
	• **Received Volume.** The number of KBs per second received by the server from clients.
	• **Sent Volume.** The number of KBs per second sent by the server to clients.
Where to go next	Clicking a chart displays a pop-up that provides additional, related information.

Workload Information

The Workload Information pane for DB2 databases v9.7.0.1 and later displays a Request Time Breakdown chart for the selected partition. The Workload Information pane for DB2 databases earlier than v9.7.0.1 displays an Average Active Sessions chart for the selected partition. The pane is located in the upper middle of the Partition (History) home page.

Description of the Pane

Data displayed for DB2 databases v9.7.0.1 and later	• **Request Time Breakdown.** The request time breakdown, for a particular partition criterion, plotted over the course of the selected time range. Click the arrow at the right of "Breakdown" to choose a different criterion from the list.
Data displayed for DB2 databases earlier than v9.7.0.1	• **Average Active Sessions.** A comparison of the average number of active sessions over the course of the selected time range with the average number of active sessions expected (according to the baseline).
Where to go next	Mousing over a data point on the chart displays a pop-up that provides more detailed information about the data point.
	Clicking the icon next to "Workload" takes you to the Current Agents pane. For more information, see Current Agents on page 67.

HADR

If the database is configured for HADR, the HADR pane displays the associated HADR metrics for the partition. It is located in the middle of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Role.** The HADR role of the database.
	• **HADR Remote Host.** The name of the HADR remote host.
	• **HADR Remote Instance.** The name of the HADR remote instance.
	• **Log Gap.** The size (in KB) of the gap between the primary log sequence number and the standby log sequence number.
	• **HADR State.** The HADR state of the database:
	▪ Disconnected
	▪ Performing local catch-up
	▪ Waiting to connect to its partner to perform remote catch-up
	▪ Performing remote catch-up
	▪ In a peer state
Where to go next	Clicking the icon next to a metric displays a pop-up with additional, related information.

For more information on these parameters, see

Process Activity

The Process Activity pane displays agent and SQL information for the selected partition. It is at the lower middle of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Agents**. The number of agents for the applications currently running in the partition.
	• **Top Agents.** The maximum number of agents for all applications running in the partition.
	• **Transactions/s.** The number of transactions involving the partition that were being carried out per second, during the last collection.
	• **Select Stmt.** The number of static SQL statements that were being run per second by the partition, during the last collection.
	• **DDL Stmt.** The number of data definition language (DDL) SQL statements that were being run per second by the partition, during the last collection.
Where to go next	Clicking the icon next to a metric displays a pop-up with additional, related information.

Memory Activity

The Memory Activity pane displays memory-related metrics for the selected partition. It is located in the middle of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Memory Total (MB).** The total memory available for the partition.
	• **Memory Used.** The percentage of the total memory available that was used by the database partition during the last collection.
	• **Buffer Pool Total (MB).** The total buffer pool available for the partition.
	• **Buffer Pool Hit Ratio.** The percentage of database page I/O requests that are being satisfied using the buffer cache (and therefore the database does not have to perform disk reads).
	• **Catalog Cache Total (MB).** The total catalog cache available for the partition.
	• **Catalog Cache Hit Ratio.** The number of times that a table descriptor or authorization information was not found in the catalog cache and had to be inserted there. Calculated as a percentage of the total number of times that a table descriptor or authorization information was requested, over the course of the selected time range.
	• **Package Cache Total (MB).** The total package cache available for the partition.
	• **Package Cache Hit Ratio.** The number of times that a requested section was not available for use and had to be loaded into the package cache. Calculated as a percentage of the total number of times that a section was requested, over the course of the selected time range.
Where to go next	Clicking a chart displays a pop-up that provides additional, related information.

I/O Activity

The I/O activity for the selected partition is displayed in charts on either side of the Memory Activity pane.

Description

Data displayed	• **Logical Reads.** The number of logical read requests that were being processed per second by the partition, during the last collection.
	• **Log I/O Reads.** The number of logical I/O read requests that were being processed per second by the partition, during the last collection.
	• **Log I/O Writes.** The number of logical I/O write operations that were being run per second by the partition, during the last collection.
	• **Async I/O Reads.** The number of asynchronous I/O read requests that were being processed per second by the partition, during the last collection.
	• **Async I/O Writes.** The number of asynchronous I/O write operations that were being run per second by the partition, during the last collection.
	• **Sync I/O Reads.** The number of synchronous I/O read requests that were being processed per second by the partition, during the last collection.
	• **Sync I/O Writes.** The number of synchronous I/O write operations that were being run per second by the partition, during the last collection.
Where to go next	Clicking a chart displays a pop-up that provides additional, related information.

Active Log

The Active Log pane displays metrics for the selected partition's active log. It is at the upper right of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Total Utilization.** The total log space that is available to the partition and the log space that is still free to be used by the partition.
	• **Log buffer full.** The total number of times that agents, that copy log records into the log buffer, have had to wait for log data to be written to disk. This value is incremented once for each time an agent experiences such contention. If two agents are waiting at the same time, this value is incremented by two.
	• **Secondaries.** The total size of the secondary files (in KBs.)
	• **Log to recovery.** The size of the recovery log (in KBs).
	• **Appl id oldest Xact.** The ID of the application holding the oldest transaction.
Where to go next	Clicking the icon below the chart displays a pop-up with additional, related information.

Disk Storage

The Disk Storage pane displays the disk storage status for the selected partition. It is at the bottom right of the Partition (Real Time) home page.

Description of the Pane

Data displayed	• **Fixed Storage.** The total disk space that is available for use by fixed-size tablespaces and the disk space that is still free to be used by fixed-size tablespaces.
	• **Auto Storage.** The total file system space that is available for the partition to use to carry out auto storage tablespace operations, and the file system space that is still free for the partition to use to carry out auto storage tablespace operations.
	• **Max TBSP Utilization.** The maximum tablespace utilization of the system.
Where to go next	Clicking either of the charts displays a pop-up with additional, related information.

Collections and Metrics

The Foglight *for DB2 LUW* agents collect metrics and store them in the Foglight topology model as topology objects.

CF Locks Collection

Tracks how database members use DB2 pureScale cluster caching facility (CF) locks.

Metric	Description
DATABASE	The name of the monitored database.
MEMBER	The name of the member.
CF_RESPONSE_TIME	Average interconnect network time per CF call calculated over all CF calls.
CF_WAIT_TIME	Time (in milliseconds) spent communicating with the cluster caching facility. Both network time and processing time. This time does not include time spent on granting locks or performing page reclaims.
CF_WAITS	Number of times the database system waited while communicating with the cluster caching facility. This is approximately the number of CF calls.
LOCK_ESCALS	Number of times that locks were escalated from several row locks to a table lock.
LOCK_ESCALS_GLOBAL	Number of lock escalations on a global lock due to global lock memory usage reaching the limit specified in the cf_lock_sz database configuration parameter.
LOCK_TIMEOUTS	Total number of lock timeouts.
LOCK_TIMEOUTS_GLOBAL	Number of lock timeouts where the application holding the lock was on a remote member
LOCK_WAIT_TIME	Total time spent waiting for locks.
LOCK_WAIT_TIME_GLOBAL	Time spent waiting for global locks in milliseconds.
LOCK_WAITS	Total number of locks waits.
LOCK_WAITS_GLOBAL	Number of locks waits due to the application holding the lock on a remote member.
GLOBAL_ESCALS_PERCENTAGE	= LOCK_ESCALS_GLOBAL / LOCK_ESCALS
GLOBAL_LOCK_TIMEOUT_PERCENTAGE	= LOCK_TIMEOUTS_GLOBAL / LOCK_TIMEOUTS
GLOBAL_LOCK_WAIT_TIME_PERCENTAGE	= LOCK_WAIT_TIME_GLOBAL / LOCK_WAIT_TIME
GLOBAL_LOCKS_PERCENTAGE	= LOCK_WAITS_GLOBAL / LOCK_WAITS.

CF Memory Pools Collection

Tracks how hosts use DB2 pureScale cluster caching facility (CF) memory pools.

Metric	Description
HOST_NAME	The name of the monitored host.
HOSTID	The ID of the CF component.
CF_LOCK_MEM_UTILIZATION	CF Lock Pool Utilization
CF_MEM_UTILIZATION	CF Memory Utilization
CF_SCA_MEM_UTILIZATION	CF SCA Pool Utilization
CONFIGURED_CF_GBP_SIZE	Configured Global Buffer Pool Size
CONFIGURED_CF_LOCK_SIZE	Configured Lock Memory Size
CONFIGURED_CF_MEM_SIZE	Configured CF Memory Size
CONFIGURED_CF_SCA_SIZE	Configured SCA Memory Size
CURRENT_CF_GBP_SIZE	Current Global Buffer Pool Size
CURRENT_CF_LOCK_SIZE	Current Lock Memory Size
CURRENT_CF_MEM_SIZE	Current Total CF Memory Size
CURRENT_CF_NONALLOCATED _SIZE	Current amount of memory allocated for the CF but not used for the GBP, LOCK, or SCA pools.
CURRENT_CF_SCA_SIZE	Current SCA Memory Size
GBP_UTILIZATION	Global Buffer Pool Utilization

CF Resource Usage Collection

Tracks how instances use DB2 pureScale cluster caching facility (CF) resources.

Metric	Description
INSTANCE	The name of the monitored instance.
ID	The ID of the CF component.
HOST	The name of the machine that hosts the CF component.
CPU_USAGE_TOTAL	Percentage of overall CPU usage of the CF machine.
MEMORY_FREE	Amount of free physical memory (MB).
MEMORY_SWAP_FREE	Amount of free swap space (MB).
MEMORY_SWAP_TOTAL	Total amount of swap space (MB).
MEMORY_TOTAL	Total size of physical memory (MB).
MEMORY_UTILIZATION	The percentage of free memory out of the total memory assigned to the CF machine.
VIRTUAL_MEMORY_FREE	Amount of reserved virtual memory.
VIRTUAL_MEMORY_TOTAL	Total amount of virtual memory on the system (MB).

Current Agents Details Collection

The following table provides a list of the Current Agents Details metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
AGENT_ID	The unique ID for the application. On a single-partitioned database, this ID consists of a 16-bit counter. On a multi-partitioned database, this ID consists of the coordinating partition number concatenated with a 16-bit counter. In addition, this ID is the same on every partition where the application may make a secondary connection.
APPL_CON_TIME	The date and time that an application started a connection request.
APPL_NAME	The name of the application running at the client.
APPL_ID	This ID is generated when the application connects to the database. It is unique across the network. There are different formats for the application ID, which depend on the communication protocol between the client and the server machine on which the database manager resides.

Metric	Description
APPL_STATUS	The status of the application. This element can help you diagnose potential application problems. Values: BACKUP COMMIT_ACT COMP CONNECTED CONNECTPEND CREATE_DB DECOUPLED DISCONNECTPEND INTR IOERROR_WAIT LOAD LOCKWAIT QUIESCE_TABLESPACE RECOMP REMOTE_RQST RESTART RESTORE ROLLBACK_ACT ROLLBACK_TO_SAVEPOINT TEND THABRT THCOMT TPREP UNLOAD UOWEXEC UOWWAIT WAITFOR_REMOTE
NUM_ASSOC_AGENTS	The number of subagents associated with an application.
COORD_NODE_NUM	In a multi-node system, the node number of the node where the application is connected or attached to the instance.
ACTIVE_TIME	The elapsed time (in seconds) spent processing an SQL request on a host database server.
EXECUTION_ID	The ID that the user specified when logging in to the operating system. This ID is distinct from AUTH_ID, which the user specifies when connecting to the database.

Metric	Description
AUTHORITY_LVL	This interface returns a text identifier based on the database authorities defined in *sql.h* and their source.
	Possible "authority" values:
	BINDADD
	CONNECT
	CREATE_EXT_RT
	CREATE_NOT_FENC
	CREATETAB
	DBADM
	IMPLICIT_SCHEMA
	LOAD
	LIBADM
	QUIESCE_CONN
	SECADM
	SYSADM
	SYSCTRL
	SYSMAINT
	SYSMON
	SYSQUIESCE
	Possible "source" values:
	USER — authority granted to the user or to a role granted to the user
	GROUP — authority granted to a group to which the user belongs or to a role granted to the group to which the user belongs
CLIENT_PID	The process ID of the client application that made the connection to the database.
COORD_AGENT_PID	The ID of the coordinator agent for the application.
STATUS_CHANGE_TIME	The date and time the application entered its status.
CLIENT_PLATFORM	This interface returns a text identifier based on the client platform definition in *sqlmon.h*.
CLIENT_PROTOCOL	This interface returns a text identifier based on the definition in *sqlmon.h*.
	Values:
	CPIC
	LOCAL
	NETBIOS
	NPIPE
	TCPIP (for DB2® UDB)
	TCPIP4
	TCPIP6
TERRITORY_CODE	The territory code of the database for which the monitor data is collected.
CODEPAGE_ID	The code page at the partition where the monitored application started.
PRIMARY_AUTH_ID	The authorization ID of the user who invoked the application that is being monitored.

Metric	Description
SESSION_AUTH_ID	The current authorization ID for the session used by this application.
CLIENT_NNAME	Deprecated by IBM.
CLIENT_DB_ALIAS	The alias of the database, provided by the application, to connect to the database.
DB_PATH	The full path of the location where the database is stored on the monitored system.
STMT_TEXT	The SQL statement text.
FOREGROUND_BACKGROUND_ TYPE	The foreground or background status of the application.
ROWS_READ	This is the number of rows read from the table.
ROWS_WRITTEN	The number of rows changed (inserted, deleted, or updated) in the table.
APPL_IDLE_TIME	The amount of time (in seconds) that has passed since an application has issued any requests to the server. Includes applications that have not terminated a transaction (for example, not issued a commit or rollback).
POOL_DATA_L_READS	The number of data pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_INDEX_L_READS	The number of index pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_INDEX_P_READS	The number of index pages read in from the tablespace containers (physical) for regular and large tablespaces.
POOL_TEMP_DATA_L_READS	The number of data pages that have been requested from the buffer pool (logical) for temporary tablespaces.
POOL_TEMP_DATA_P_READS	The number of data pages read in from the tablespace containers (physical) for temporary tablespaces.
POOL_TEMP_INDEX_L_READS	The number of index pages that have been requested from the buffer pool (logical) for temporary tablespaces.
POOL_TEMP_INDEX_P_READ S	The number of index pages read in from the tablespace containers (physical) for temporary tablespaces.
LAST_RESET	The date and time when the monitoring counters were reset.
PACKAGE_CACHE_HIT_RATIO	A ratio that indicates how well the package cache is helping to avoid the reload of packages and sections for static SQL from the system catalogs and the recompilation of dynamic SQL statements. The indicator is calculated using the formula: (1-(db.pkg_cache_inserts/db.pkg_cache_lookups))*100

Agents Summary Collection

The following table provides a list of the Agents Summary metrics that are collected, and a description for each.

Metric	Description
ACTIVE	The number of active agents.
ACTIVE_Rate	The number of active agents per second (average active sessions).
BACKGROUND_PROCESSES	The sum of background application agents.
BACKUP_AGENTS	Indicates that the application is performing a backup of the database.
BALANCE_AVERAGE_ACTIVE_ SESSIONS	The balance of average active agents across the database partitions. If most of the activity is on one partition, for example, the activity is not balanced. If most of the activity is spread equally across the partitions, it is balanced.
COMMIT_ACT_AGENTS	Indicates that the unit of work is committing its database changes.
COMP_AGENTS	Indicates that the database manager is compiling an SQL statement or precompiling a plan on behalf of the application.
CONNECTPEND_AGENTS	Indicates that the application has initiated a database connection, but the request has not yet completed.
CREATE_DB_AGENTS	Indicates that the agent has initiated a request to create a database, but that request has not yet completed.
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
DECOUPLED_AGENTS	Indicates that there are no agents currently associated with the application. This is a normal state. When the Connection Concentrator is enabled, there is no dedicated coordinator agent, so an application can be decoupled on the coordinator partition. In non-concentrator environments, an application cannot be decoupled on the coordinator partition, because there is always a dedicated coordinator agent.
DISCONNECTPEND_AGENTS	Indicates that the application has initiated disconnection from the database, but the command has not yet completed running. The application may not have explicitly run the database disconnect command. The database manager disconnects from a database if the application ends without disconnecting.
FOREGROUND_PROCESSES	The sum of foreground application agents.
INACTIVE	Indicates that the application is in one of three states: connected, uowwait, or decoupled.
INTR_AGENTS	Indicates that an interrupt of a request is in progress.
IOERROR_WAIT_AGENTS	Indicates that the application has detected an I/O error and is attempting to disable a particular tablespace. The application has to wait for all other active transactions on the tablespace to complete before it can disable the tablespace.
LOAD_AGENTS	Indicates that the application is performing a "fast load" of data into the database.
LOCKWAIT_AGENTS	Indicates that the unit of work is waiting for a lock. After the lock is granted, the status is restored to its previous value.
QUIESCE_TABLESPACE_AGEN TS	Indicates that the application is performing a quiesce tablespace request.

Metric	Description
RECOMP_AGENTS	Indicates that the database manager is recompiling (that is, rebinding) a plan on behalf of the application.
REMOTE_RQST_AGENTS	Indicates that the application is waiting for results from a federated data source.
RESTART_AGENTS	Indicates that the application is restarting a database in order to perform crash recovery.
RESTORE_AGENTS	Indicates that the application is restoring a backup image to the database.
ROLLBACK_ACT_AGENTS	Indicates that the unit of work is rolling back its database changes.
ROLLBACK_TO_SAVEPOINT_AGENTS	Indicates that the application is rolling back to a savepoint.
TEND_AGENTS	Indicates that the unit of work is part of a global transaction that has ended, but has not yet entered the prepared phase of the two-phase commit protocol.
THABRT_AGENTS	Indicates that the unit of work is part of a global transaction that has been heuristically rolled back.
THCOMT_AGENTS	Indicates that the unit of work is part of a global transaction that has been heuristically committed.
TOTAL_AGENTS	The total number of agents of all types.
TOTAL_AGENTS_Rate	The rate of total agents per second.
UNLOAD_AGENTS	Indicates that the application is performing a "fast unload" of data from the database.
UOWEXEC_AGENTS	Indicates that the database manager is running requests on behalf of the unit of work.
UOWWAIT_AGENTS	Indicates that the database manager is waiting on behalf of the unit of work in the application. This status typically means that the system is running in the application's code.
WAITFOR_REMOTE_AGENTS	Indicates that the application is waiting for a response from a remote partition in a partitioned database instance.
OTHER	The agents not identified by one of the agent types we show at the workload agents.

Current Locks Wait Collection

The following table provides a list of the Current Locks Wait metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
LOCK_NAME	The ID of the lock.

Metric	Description
AGENT_ID	A system-wide unique ID for the locked wait application. On a single-partitioned database, this ID consists of a 16-bit counter. On a multi-partitioned database, this ID consists of the coordinating partition number concatenated with a 16-bit counter. In addition, this ID is the same on every partition where the application may make a secondary connection.

The application handle can be used to uniquely identify an active application (application handle is synonymous with agent ID). It can also be used as input to the FORCE APPLICATION command or API. On multi-node systems, this command can be issued from any node where the application has a connection. Its effect is global. |
| STMT_TEXT | The full statement text of the waited (locked) agent. |
| LOCK_MODE | The type of lock that is being held.

This mode can help in determining the source of the contention for resources.

This element indicates one of the following, depending on the type of monitoring information being examined:

- The type of lock another application holds on the object that this application is waiting to lock (for application-monitoring and deadlock-monitoring levels).
- The type of lock held on the object by this application (for object-lock levels).

Mode Types:

NON — No Lock

IS — Intention Share Lock

IX — Intention Exclusive Lock

S — Share Lock

SIX — Share with Intention Exclusive Lock

X — Exclusive Lock

IN — Intent None

Z — Super Exclusive Lock

U — Update Lock

NS — Next Key Share Lock

NX — Next Key Exclusive Lock

W — Weak Exclusive Lock

NW — Next Key Weak Exclusive Lock |

Metric	Description
LOCK_OBJECT_TYPE	The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information). This element can help in determining the source of the contention for resources. The object type IDs are defined in *sqlmon.h*. The objects may be one of the following types: • Tablespace • Table • Buffer pool • Block • Record (or row) • Data Partition • Internal (another type of lock held internally by the database manager) • Automatic resize • Automatic storage • Values: • AUTORESIZE_LOCK • AUTOSTORAGE_LOCK • BLOCK_LOCK • EOT_LOCK • INPLACE_REORG_LOCK INTERNAL_LOCK • INTERNALB_LOCK • INTERNALC_LOCK • INTERNALJ_LOCK • INTERNALL_LOCK • INTERNALO_LOCK • INTERNALQ_LOCK • INTERNALP_LOCK • INTERNALS_LOCK • INTERNALT_LOCK • INTERNALV_LOCK • KEYVALUE_LOCK • ROW_LOCK • SYSBOOT_LOCK • TABLE_LOCK • TABLE_PART_LOCK • TABLESPACE_LOCK • XML_PATH_LOCK
AGENT_ID_HOLDING_LK	Agent ID of the blocker that holds the object or the application handle of the agent holding a lock for which this application is waiting. The lock monitor group must be turned on to obtain this information. This element can help in determining which applications are in contention for resources. If this element is 0 (zero) and the application is waiting for a lock, indicates that the lock is held by an in-doubt transaction.

Metric	Description
APPL_ID_HOLDING_LK	The application ID of the application that is holding a lock on the object that this application is waiting to obtain.
LOCK_WAIT_START_TIME	The date and time that this application started waiting to obtain a lock on the object that is locked by another application. This element can help in determining the severity of resource contention.
LOCK_MODE_REQUESTED	The lock mode requested by the application. Mode Types: NON — No Lock IS — Intention Share Lock IX — Intention Exclusive Lock S — Share Lock SIX — Share with Intention Exclusive Lock X — Exclusive Lock IN — Intent None Z — Super Exclusive Lock U — Update Lock NS — Next Key Share Lock NX — Next Key Exclusive Lock W — Weak Exclusive Lock NW — Next Key Weak Exclusive Lock
LOCK_ESCALATION	Indicates whether a lock request was made as part of a lock escalation. Use this element to understand the cause of deadlocks. If you experience a deadlock that involves applications doing lock escalation, you may want to increase the amount of lock memory or change the percentage of locks that any one application can request.
DATA_PARTITION_ID	The ID of the data partition for which a lock occurred. This element is only applicable to partitioned tables. A value of -1 represents a lock of the whole table and not a specific partition.
TABNAME	The name of the table. Along with TABSCHEMA, this element can help in determining the source of the contention for resources. At the application level, application-lock level, and deadlock-monitoring level, this is the table that the application is waiting to lock, because it is locked by another application. For snapshot monitoring, this item is valid only when the "lock" monitor group information is turned on, and when LOCK_OBJECT_TYPE indicates that the application is waiting to obtain a table lock.
TABSCHEMA	The schema of the table. Along with TABNAME, this element can help in determining the source of the contention for resources.

Metric	Description
TBSP_NAME	The name of a tablespace.
	This element can help in determining the source of the contention for resources.
	It is equivalent to the TBSPACE column in the database catalog table SYSCAT.TABLESPACES.
	• At the application level, application-lock level, and deadlock-monitoring level, this is the name of the tablespace that the application is waiting to lock. Another application currently holds a lock on this tablespace.
	• At the lock level, this is the name of the tablespace against which the application currently holds a lock.
	• At the tablespace level (when the buffer pool monitor group is ON), this is the name of the tablespace for which information is returned.
	This element is not returned for a table lock held on a partitioned table.
LOCK_ATTRIBUTES	The lock attributes of the lock wait. If there are no locks, the text identifier is NONE; otherwise, it is any combination of the following, separated by a '+' sign:
	ALLOW_NEW
	DELETE_IN_BLOCK
	ESCALATED
	INSERT
	NEW_REQUEST
	RR
	RR_IN_BLOCK
	UPDATE_DELETE
	WAIT_FOR_AVAIL
LOCK_CURRENT_MODE	During a lock conversion operation, the type of lock held before the conversion is completed. The following is an example of a scenario that describes lock conversion: During an update or delete operation, it is possible to wait for an X lock on the target row. If the transaction is holding an S or V lock on the row, this would require a conversion. The LOCK_CURRENT_MODE element is assigned a value of S or V, while the lock waits to be converted to an X lock.
LOCK_RELEASE_FLAGS	The lock release flags.
	Each release flag is based on a bit flag value defined in *sqlmon.h*. The following are possible release flag settings (API Constant Description):
	SQLM_LOCKRELFLAGS_SQLCOMPILER — Locks by SQL compiler
	SQLM_LOCKRELFLAGS_UNTRACKED — Non-unique untracked locks
SUBSECTION_NUMBER	Identifies the subsection associated with the returned information.
	This number relates to the subsection number in the access plan that can be obtained with db2expln.
LOCK_DURATION	The duration of the lock wait.

Database Bufferpools Summary PS Collection

Tracks how members use group buffer pools (GBP) and local buffer pools (LBP).

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
MEMBER	The name of the database member from which the data was retrieved.
GBP_EFFICENCY	Percentage of pages read from the GBP after the page found in the LBP was found to be invalid. Indicates the amount of time a read from disk was avoided thanks to the GBP.
GBP_HIT_RATIO	*pureScale only* — Percentage of times a requested page was found in the GBP out of all times it was requested. If a page is found in the GBP, it saves the database the time to look for it in the LBP.
GBP_INVALID_PAGES	Number of GBP data page read attempts due to an LBP page being present but marked invalid.
GBP_LOGICAL_READS	*pureScale only* — Number of times a GBP-dependent page was attempted to be read from the GBP because the page was either invalid or not present in the LBP.
GBP_PHYSICAL_READS	Number of times a GBP-dependent page was read into the LBP from disk because it was not found in the GBP.
LBP_HIT_RATIO	Percentage of times a requested page was found in the LBP (either valid or invalid) out of all times it was requested.
LBP_LOGICAL_READS	Number of pages (Data, Index, XDA) that were found in the LBP. Value includes both valid and invalid pages.
LBP_PHYSICAL_READS	Number of pages prefetched from the GBP to the LBP.
TOTAL_HIT_RATIO	Percentage of times a requested page was found in the buffer pool out of all times it was requested. If a page is found in the buffer pool, it saves the database the time to look for it in the physical storage.
TOTAL_LOGICAL_READS	Total logical reads (index, XDA and data) requested from the buffer pool.
TOTAL_PHYSICAL_READS	Total pages (index, XDA and data) read from physical storage into the buffer pool.

Database Configuration Collection

The following table provides a list of the Database Configuration metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
HADR_DB_ROLE	Standard means that the database is not a HADR DB. Primary or Standby means that it is.
LOGFILESIZE	A parameter that predefines the size of the diag log. If this is set to 0 (the default), the diag log gets larger with no limit. If a size is specified, after the diag log reaches that size, a new file with a new name is created and used.

Database Tablespaces Collection

The following table provides a list of the Database Tablespaces metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
TBSP_NAME	The tablespace name.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
TABLESPACE_ID	The tablespace ID.
TABLESPACE_STATE	This interface returns a text identifier based on the definitions in *sqlutil.h*, and is a combination of the following values, separated by a '+' sign: BACKUP_IN_PROGRESS BACKUP_PENDING DELETE_PENDING DISABLE_PENDING DROP_PENDING LOAD_IN_PROGRESS LOAD_PENDING NORMAL OFFLINE PSTAT_CREATION PSTAT_DELETION QUIESCED_EXCLUSIVE QUIESCED_SHARE QUIESCED_UPDATE REBAL_IN_PROGRESS REORG_IN_PROGRESS RESTORE_IN_PROGRESS RESTORE_PENDING ROLLFORWARD_IN_PROGRESS ROLLFORWARD_PENDING STORDEF_ALLOWED STORDEF_CHANGED STORDEF_FINAL_VERSION STORDEF_PENDING SUSPEND_WRITE
TABLESPACE_USING_AUTO_STORAGE	Specifies whether the tablespace was created as an automatic storage tablespace. A value of 1 indicates yes; a value of 0 indicates no.
TABLESPACE_AUTO_RESIZE_ENABLED	This element is applicable to DMS tablespaces and non-temporary automatic storage tablespaces only. If this element is set to 1, then automatic resizing is enabled.

Metric	Description
TABLESPACE_TYPE	This interface returns a text identifier, based on the definitions in *sqlutil.h*, which is one of the following values: DMS SMS
TABLESPACE_FREE_SIZE_MB	The tablespace free size [MB].
TABLESPACE_TOTAL_SIZE_MB	The tablespace total size [MB].
TABLESPACE_USED_SIZE_MB	The tablespace used size [MB].
TABLESPACE_USABLE_SIZE_MB	The tablespace usable size [MB].
TABLESPACE_UTILIZATION_ PERCENT	The percentage of tablespace used.
TABLESPACE_PAGE_SIZE_KB	The tablespace page size (in KB).
TABLESPACE_TOTAL_PAGES	The total operating system space occupied by a tablespace. For DMS, this is the sum of the container sizes (including overhead). For SMS, this is the sum of all the filespace used for the tables stored in this tablespace (and is collected only if the buffer pool switch is on).
TABLESPACE_USABLE_PAGES	The total number of pages in a tablespace minus the overhead pages. This element is applicable to DMS tablespaces only. For SMS, this element has the same value as TABLESPACE_TOTAL_PAGES.
TABLESPACE_FREE_PAGES	The total number of pages that are currently free in a tablespace. This is applicable to DMS tablespaces only.
TABLESPACE_USED_PAGES	The total number of pages in use for a DMS tablespace. For an SMS tablespace, this is equal to TABLESPACE_TOTAL_PAGES.
TABLESPACE_PREFETCH_SIZE	The maximum number of pages the prefetcher gets from the disk at a time. If automatic prefetch size is enabled, this element reports the value -1.
TABLESPACE_EXTENT_SIZE	The tablespace extent size.
TABLESPACE_NUM_CONTAINERS	The number of containers.
FILES_CLOSED	The total number of closed database files. The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files open by an application at any time is controlled by the maxfilop configuration parameter.
TABLESPACE_CONTENT_TYPE	This interface returns a text identifier, based on the definitions in *sqlmon.h*, which is one of the following values: ANY LARGE SYSTEMP USRTEMP

Metric	Description
TABLESPACE_REBALANCER_MODE	Specifies whether a forward or reverse rebalance is taking place. The potential values (defined in *sqlmon.h*) are: SQLM_TABLESPACE_NO_REBAL — no rebalancing taking place SQLM_TABLESPACE_FWD_REBAL — forward SQLM_TABLESPACE_REV_REBAL — reverse This can be used to indicate whether the current rebalance process is removing space from a tablespace or adding space to a tablespace. This is only applicable to a DMS tablespace.
TABLESPACE_LAST_RESIZE_TIME	This element displays a timestamp that represents the last time the size of the tablespace was successfully increased. For tablespaces that can be automatically resized, this element represents the last time that space was automatically added to the tablespace when it became full, more space was being requested, and the maximum tablespace size had not been reached. This element is only applicable to tablespaces that are enabled to be automatically resized.
TABLESPACE_LAST_RESIZE_FAILED	This element describes whether the last attempt to automatically increase the size of the tablespace failed. A value of 1 indicates yes, and a value of 0 indicates no. For an automatic storage tablespace, this element may show that there is no space left on any of the database's storage paths. For a non-automatic storage tablespace, a failure means that one of the containers could not be extended because its filesystem was full. Another reason for failure is that the maximum size of the tablespace has been reached. This element is only applicable to tablespaces that are enabled to be automatically resized.
TBSP_CURRENT_SIZE_MB	The current size of DMS tablespace (MB). For DMS and automatic storage tablespaces, this element represents the total size of all tablespace containers in bytes. This value is equal to the total pages for the tablespace (TABLESPACE_TOTAL_PAGES) multiplied by the tablespace's page size (TABLESPACE_PAGE_SIZE). This element is not applicable for SMS tablespaces or for temporary automatic storage tablespaces. On tablespace creation, for an automatic storage tablespace, the current size might not match the initial size. The value of current size will be within page size multiplied by extent size multiplied by the number of storage paths of the initial size on creation (usually greater, but sometimes smaller). It will always be less than or equal to TABLESPACE_MAX_SIZE (if set). This is because containers can only grow by full extents, and must be grown as a set.

Metric	Description
TBSP_MAX_SIZE_MB	The tablespace max size of DMS tablespace (MB). This element shows the maximum size to which the tablespace can automatically resize or increase.
	This represents the maximum size in bytes to which a tablespace that can be automatically resized can automatically increase. If this value is equal to the TABLESPACE_CURRENT_SIZE element, then there is no room for the tablespace to grow. If the value of this element is -1, then the maximum size is considered to be "unlimited", and the tablespace can automatically resize until the file systems are full or the architectural size limit of the tablespace is reached. (This limit is described in the SQL Limits appendix of the SQL Reference). This element is only applicable to tablespaces that are enabled for automatic resizing.
TABLESPACE_DMS_RESIZE_ UTILIZATION	The percentage of tablespace DMS *auto-resize yes* and *auto-storage no* used.

Database Tablespaces Summary Collection

The following table provides a list of the Database Tablespaces Summary metrics that are collected, and a description for each.

Metric	Description
OVERALL_FIXED_TABLESPACE_ TOTAL_SIZE_MB	The total size [MB] of fixed DMS tablespaces.
OVERALL_FIXED_TABLESPACE_ USED_SIZE_MB	The overall used size [MB] of fixed DMS tablespace.
OVERALL_FIXED_TABLESPACE_ UTILIZATION	The overall utilization (%) of fixed DMS tablespaces.
OVERALL_AUTO_TABLESPACE _ USED_SIZE_MB	The tablespace total size [MB].
OVERALL_AUTO_TABLESPACE _ UTILIZATION	The overall utilization (%) of AUTO DMS and SMS tablespaces spread across different file systems and hosts used by DB2.
AUTO_STORAGE_STATE	Flag returns 0, 1, or 0, 1. 0 means Fixed 1 means Auto 0, 1 means both

Database Usability Collection

The following table provides a list of the Database Usability metrics that are collected, and a description for each.

Metric	Description
DB_RESPONSE_TIME	The response time of a query representing the general workload ("select 1 from sysibm.sysdummy1") when the connection is already open.
DB_CONNECTION_TIME	The time for establishing the connection.
DB_CONNECTION_AVAILABILITY	The percent of nodes for which the connection is available.

Database Parameters Collection

The following table describes the metrics that are collected for each database configuration parameter.

Metric	Description
DATABASE_NAME	The name of the monitored database.
NAME	The configuration parameter name.
CURRENT_VALUE	The current value of the configuration parameter stored in memory, in partition/member 0.
VALUE_FLAGS	Provides information about the current value of the configuration parameter. Values: NONE — No additional information AUTOMATIC — The configuration parameter has been set to automatic
DEFERRED_VALUE	The value of the configuration parameter on disk. For some database configuration parameters, changes only take effect when the database is reactivated. In such cases, all applications must first disconnect from the database. (If the database was enabled, then it must be deactivated and then reactivated.) The changes take effect with the next connection to the database.
DATATYPE	The configuration parameter data type.
MODIFIED	Indicates whether the value changed since the last collection. If the value changed on disk, but the instance is still using the previous value, the value of this parameter is 0.

Database Instance Parameters Collection

The following table describes the metrics that are collected for each database instance configuration parameter.

Metric	Description
INSTANCE_NAME	The database instance name.
NAME	The configuration parameter name.
CURRENT_VALUE	The current value of the configuration parameter stored in memory, in partition/member 0.

Metric	Description
VALUE_FLAGS	Provides specific information about the current value of the configuration parameter. Values: NONE — No additional information AUTOMATIC — The configuration parameter has been set to automatic
DEFERRED_VALUE	The value of the configuration parameter on disk. For some database configuration parameters, changes only take effect when the database is reactivated. In such cases, all applications must first disconnect from the database. (If the database was enabled, then it must be deactivated and then reactivated.) The changes take effect with the next connection to the database.
DATATYPE	The configuration parameter data type.
MODIFIED	Indicates whether the value changed since last collection. If the value changed on disk, but the instance is still using the previous value, the value of this parameter is 0.

Database Instance Registry Variables Collection

The following table describes the metrics that are collected for each database instance registry variable.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
REG_VAR_NAME	The name of the DB2 registry variable.
REG_VAR_VALUE	The current value of the DB2 registry variable in memory.
REG_VAR_ON_DISK_VALUE	The current setting of the DB2 registry variable on disk. For DB2 versions prior to version 10, the current setting is not available, so the value of this parameter is "N/A".
LEVEL	Indicates the level at which the DB2 registry variable acquires its value. The possible return values and the corresponding levels that they represent are: I — Instance G — Global N — Database partition E — Environment
MODIFIED	Indicates whether the value changed since last collection. If the value changed on disk, but the instance is still using the previous value, the value of this parameter is 0.

Database Tablespaces BP Collection

The following table provides a list of the Database Tablespaces BP metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
TBSP_NAME	The tablespace name.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
TABLESPACE_TYPE	This interface returns a text identifier, based on the definitions in *sqlutil.h*, which is one of the following values: DMS SMS
TABLESPACE_ID	The tablespace ID.
POOL_ASYNC_DATA_READS	The number of data pages read in from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces.
POOL_ASYNC_DATA_WRITES	The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.
POOL_ASYNC_INDEX_READS	The number of index pages read in from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces.
POOL_ASYNC_INDEX_WRITES	The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.
POOL_ASYNC_DATA_READ_REQS	The number of read requests made by the prefetcher to the operating system. These requests are typically large block IOs of multiple pages.
POOL_ASYNC_READ_TIME	The total amount of time spent reading in data and index pages from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces. This value is given in milliseconds.
POOL_ASYNC_WRITE_TIME	The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
POOL_DATA_L_READS	The number of data pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_DATA_P_READS	The number of data pages read in from the tablespace containers (physical) for regular and large tablespaces.
POOL_DATA_WRITES	The number of times a buffer pool data page was physically written to disk.
POOL_INDEX_L_READS	The number of index pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_INDEX_P_READS	The number of index pages read in from the tablespace containers (physical) for regular and large tablespaces.

Metric	Description
POOL_INDEX_WRITES	The number of times a buffer pool index page was physically written to disk. Like a data page, a buffer pool index page is written to disk for the following reasons: • To free a page in the buffer pool so that another page can be read • To flush the buffer pool
POOL_READ_TIME	The total amount of time spent reading in data and index pages from the tablespace containers (physical) for all types of tablespaces. This value is given in milliseconds.
POOL_WRITE_TIME	Provides the total amount of time spent physically writing data or index pages from the buffer pool to disk. The elapsed time is given in milliseconds.
DIRECT_READ_REQS	The number of requests to perform a direct read of one or more sectors of data.
DIRECT_READ_TIME	The elapsed time (in milliseconds) required to perform the direct reads.
DIRECT_READS	The number of read operations that do not use the buffer pool.
DIRECT_WRITE_REQS	The number of requests to perform a direct write of one or more sectors of data.
DIRECT_WRITE_TIME	The elapsed time (in milliseconds) required to perform the direct writes.
DIRECT_WRITES	The number of write operations that do not use the buffer pool.
INDEX_HIT_RATIO	The percentage of index reads that were carried out from the buffer pool for the selected tablespace without having to perform physical reads. 100 * (1 - (POOL_INDEX_P_READS / (POOL_INDEX_L_READS)))
BUFFER_POOL_HIT_RATIO	The total hit ratio calculation for the selected tablespace. 100 * (1 - ((POOL_DATA_P_READS + POOL_INDEX_P_READS) / (POOL_DATA_L_READS + POOL_INDEX_L_READS)))

FCM Activity Collection

The following table provides a list of the FCM Activity metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.

Metric	Description
CONNECTION_STATUS	This interface returns a text identifier, based on the definition in *sqlmon.h*, which is one of the following values: INACTIVE ACTIVE CONGESTED
CONNECTION_AVAILABILITY	The agent's connection FCM availability. Possible values are either 0 (INACTIVE or CONGESTED) or 100 (ACTIVE).
FCM_DBPARTITIONNUM	The database partition number to which data was sent or from which data was received (as per the TOTAL_BUFFERS_SENT and TOTAL_BUFFERS_RCVD columns).
TOTAL_BUFFERS_SENT	The total number of FCM buffers sent from the current node to the remote node. High values of this metric can indicate the need to redistribute the database or to move tables to reduce the inter-node traffic.
TOTAL_BUFFERS_RCVD	The total number of FCM buffers received by the node issuing the GET SNAPSHOT command, from the node identified by the NODE_NUMBER (see the db2nodes.cfg file). You can use this element to measure the level of traffic between the current node and the remote node. If the total number of FCM buffers received from this node is high, you may want to redistribute the database or move tables to reduce the inter-node traffic.
TOTAL_BUFFERS_SENT_RATE	The rate of the total number of FCM buffer pages sent from the current node to the remote node. This parameter can be used for measuring the level of traffic between the current and remote nodes. High inter-node traffic can be addressed by redistributing the database or moving tables to reduce the inter-node traffic.
TOTAL_BUFFERS_RCVD_RATE	The rate of the total number of FCM buffer pages received by the current node from the remote node. This parameter can be used for measuring the level of traffic between the current and remote nodes. High inter-node traffic can be addressed by redistributing the database or moving tables to reduce the inter-node traffic.

FS DB Storage Collection

The following table provides a list of the FS Database Storage metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
FS_ID	This element shows the unique identification number provided by the operating system for a file system pointed to by a storage path.
FS_TOTAL_SIZE	This element shows the capacity (MB) of a file system pointed to by a storage path.

Metric	Description
FS_USED_SIZE	This element shows the amount of space (MB) already used on a file system pointed to by a storage path.
STO_PATH_FREE_SIZE	This element shows the amount of free space (MB) available on a file system pointed to by a storage path. If multiple storage paths point to the same file system, the free size is not divided among them.

FS DB Storage Summary Collection

The following table provides a list of the FS Database Storage Summary metrics that are collected, and a description for each.

Metric	Description
OVERALL_FS_TOTAL_SIZE	This element shows the capacity (MB) per database of a file system pointed to by a storage path.
OVERALL_FS_USED_SIZE	This element shows the amount of space (MB) per database already used on a file system pointed to by a storage path.
OVERALL_STO_PATH_FREE_S IZE	This element shows the amount of free space (MB) per database available on a file system pointed to by a storage path. If multiple storage paths point to the same file system, the free size is not divided among them.
OVERALL_DB_FS_UTILIZATIO N	This element shows the amount of used space (MB) per database available on a file system pointed to by a storage path.

FS Instance Storage Collection

The following table provides a list of the FS Instance Storage metrics that are collected, and a description for each.

Metric	Description
OVERALL_FS_TOTAL_SIZE	The overall total space size (MB) of file systems over which the DB2 instance is spread.
OVERALL_FS_USED_SIZE	The amount of space (MB) used on the file systems over which the DB2 instance is spread (not just what DB2 is using; this metric represents the overall usage).
OVERALL_INSTANCE_FS_UTILIZ ATION	The overall utilization (%) of the file systems over which the DB2 instance is spread.
OVERALL_FREE_SIZE	This element shows the amount of free space (MB) per instance on a file system.

FS Storage Collection

The following table provides a list of the FS Storage metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
FS_ID	This element shows the unique identification number provided by the operating system for a file system pointed to by a storage path.
DB_STORAGE_PATH	This element shows the full path of a location used by the database for placing automatic storage tablespaces. There can be 0 or more storage paths associated with a database.
FS_TOTAL_SIZE	This element shows the capacity (MB) of a file system pointed to by a storage path.
FS_USED_SIZE	This element shows the amount of space (MB) already used on a file system pointed to by a storage path.
STO_PATH_FREE_SIZE	This element shows the amount of free space (MB) available on a file system pointed to by a storage path. If multiple storage paths point to the same file system, the free size is not divided among them.
FS_UTILIZATION	The utilization (%) of the file system over which the DB2 instance is spread.

FS Storage Summary Collection

The following table provides a list of the FS Storage Summary metrics that are collected, and a description for each.

Metric	Description
OVERALL_FS_TOTAL_SIZE	The total capacity (MB) per partition of the file system to which the storage path points.
OVERALL_FS_USED_SIZE	This element shows the amount of space (MB) per partition already used on a file system pointed to by a storage path.
OVERALL_STO_PATH_FREE_SIZE	This element shows the amount of free space (MB) per partition available on a file system pointed to by a storage path. If multiple storage paths point to the same file system, the free size is not divided among them.
OVERALL_FS_UTILIZATION	This element shows the amount of free space (MB) per database available on a file system pointed to by a storage path. If multiple storage paths point to the same file system, the free size is not divided among them.

General Activity Collection

The following table provides a list of the General Activity metrics that are collected, and a description for each.

Metric	Description
ACTIVE_CONNECTIONS	The number of applications that are currently connected to the database, and for which the database manager is processing a request.
ACTIVE_CONNECTIONS_RATE	The number of active connections per second.
ACTIVE_SORTS	The number of sorts in the database that have a sort heap allocated.
APPL_ID_OLDEST_XACT	The application ID of the application with the oldest transaction. **NOTE:** When a transaction is not holding up logging, or the oldest transaction does not have an application ID (for example, an in-doubt transaction or inactive transaction), the application's ID is not available.
APPL_SECTION_INSERTS	Counts the number of times a copy of a SQL executable section was not available in the application SQL work area and had to be inserted.
APPL_SECTION_LOOKUPS	Each agent has access to a unique SQL work area, where the working copy of any executable section is kept. This metric counts the total number of times the SQL work area was accessed by agents for an application. You can use this element with APPL_SECTION_INSERTS to tune the size of the SQL work area heap.
AVERAGE_LOCK_WAITER_WAIT_TIME	The average amount of time (in seconds) for a lock wait in the database.
CAT_CACHE_INSERTS	The number of times the system attempted to insert table descriptor or authorization information into the catalog cache.
CAT_CACHE_LOOKUPS	The number of times that the catalog cache was referenced to obtain table descriptor or authorization information. This includes both successful and unsuccessful attempts to access the catalog cache.
CAT_CACHE_OVERFLOWS	The number of times that the catalog cache overflowed its allocated memory. You can use this element with CAT_CACHE_SIZE_TOP to determine if the size of the catalog cache should be increased to avoid overflowing.
CAT_CACHE_SIZE_TOP	The largest size (in bytes) reached by the catalog cache including overflows, if they occurred.
CATALOG_CACHE_HIT_RATIO	A percentage that indicates how well the catalog cache is helping to avoid actual accesses to insert table descriptor or authorization information into the catalog cache, compared to the number of times the catalog cache was referenced to obtain table descriptor information or authorization information. A high ratio indicates that it is successful in avoiding actual disk I/O accesses. The indicator is calculated using the formula: (1-(CAT_CACHE_INSERTS/CAT_CACHE_LOOKUPS))*100
CATALOG_PARTITION	The main database partition in which the database catalog tables are stored.
CLIENT_PERCENT_ACTIVE	The percentage of active client applications.

Metric	Description
COMMIT_SQL_STMTS	The total number of SQL COMMIT statements attempted.
CONNECTIONS	The total number of connections that took place during the specified time range.
CONNECTIONS_HWM	The highest number of simultaneous connections to the database since the database was enabled.
CONNECTIONS_RATE	The number of connections that took place, per second, during the specified time range.
DB_CONN_TIME	The date and time of the connection to the database (at the database level, the first connection to the database), or when the enable database command was issued.
DB_LOCATION	The database location. Values: LOCAL or REMOTE At the database level, this is always LOCAL.
DB_NAME	The name of the monitored database.
DB_PATH	The full path to the database location on the monitored system. At the database level, it is the path of the coordinator partition of the database (most of the time, 0).
DB_STATUS	The status of the database. This interface returns a text identifier, based on the definition in *sqlmon.h*, which is one of the following values: ACTIVE QUIESCE_PEND QUIESCED ROLLFWD
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
DEADLOCKS	The total number of deadlocks that have occurred in the database for the required period. A deadlock occurs when there is a cyclical dependency between two or more threads or processes that are contending for the same set of resources within DB2. That is, each task has a lock on a resource that the other tasks are trying to lock.
DYNAMIC_SQL_STMTS	The number of dynamic SQL statements attempted.
DYNAMIC_SQL_STMTS_RATE	The number of dynamic SQL statements that were run, per second, during the specified time range.
ELAPSED_EXEC_TIME_S	The sum of the host execution times (in seconds) for all the statements that were run for a particular database. This element does not include the network time elapsed between DB2 Connect™ and the host database server.
ELAPSED_EXEC_TIME_S_RATE	The sum of the host elapsed time per second for all the statements that were run for a particular database.
HASH_JOIN_OVERFLOWS	The number of times that hash join data exceeded the available sort heap space.

Metric	Description
INT_COMMITS	The total number of commits initiated internally by the database manager. **NOTE:** This value does not include explicit SQL COMMIT statements.
INT_ROLLBACKS	The total number of rollbacks initiated internally by the database manager. **NOTE:** This value does not include explicit SQL COMMIT statements, but the INT_DEADLOCK_ROLLBACKS count is included.
LAST_BACKUP	The date and time that the latest database backup was completed.
LAST_RESET	The date and time that the monitor counters were last reset. If the counters have never been reset, the value of the element is zero. **NOTE:** The database manager counters are reset only if you reset all active databases.
LOCK_LIST_IN_USE_KB	The total amount of lock list memory (in KB) that is in use.
LOCK_TIMEOUTS	The number of times that a request to lock an object timed-out instead of being granted.
LOCK_WAIT_TIME_S	The total amount of time (in seconds) that all applications were waiting for a lock within this database.
LOCK_WAIT_TIME_S_RATE	The rate of time at which all applications were waiting for a lock within this database.
LOCK_WAITS	The total number of times that applications waited for locks within this database.
LOCKS_WAITING	The number of agents waiting on a lock.
NUM_ASSOC_AGENTS	The number of subagents for all applications.
PACKAGE_CACHE_HIT_RATIO	A percentage that indicates how well the package cache is performing. It is the total number of times that a requested section was not available for use and had to be loaded into the package cache compared to what was already loaded into the cache. Includes any implicit prepares performed by the system. This indicator is calculated using the formula: ((1 - (PKG_CACHE_INSERTS/ PKG_CACHE_LOOKUPS)) * 100)
PKG_CACHE_INSERTS	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This includes any implicit prepares performed by the system.
PKG_CACHE_LOOKUPS	The number of times that an application looked for a section or package in the package cache. At the database level, this is the overall number of references since the database was started, or since monitor data was reset. **NOTE:** This includes cases in which the section is already loaded in the cache and those in which the section has to be loaded into the cache.

Metric	Description
PKG_CACHE_NUM_OVERFLOWS	The number of times that the package cache overflowed its allocated memory. You can use this element with PKG_CACHE_SIZE_TOP to determine if the size of the package cache should be increased to avoid overflowing.
PKG_CACHE_SIZE_TOP	The largest size reached by the package cache. If the package cache experienced overflow, then this element contains the largest size reached by the package cache during the overflow.
POST_SHRTHRESHOLD_SORTS	The total number of sorts throttled back by the sort memory throttling algorithm. Throttled sorts are sorts that are granted less memory than requested by the sort memory manager. A sort is throttled back when the memory allocation for sorts is close to the limit set by the SHEAPTHRES_SHR database configuration parameter. Throttling significantly reduces the number SHEAPTHRES_SHR overflows in a system that is improperly configured. The value of this element only includes sorts that are using memory allocated from the shared sort heap.
ROLLBACK_SQL_STMTS	The total number of SQL ROLLBACK statements attempted. A rollback can result from an application request, a deadlock, or an error. This element only counts the rollback statements issued from applications. This metric can be used to help with determining the level of activity in the database and the amount of conflict between applications on the database.
SERVER_PLATFORM	The operating system running the database server.
SORT_HEAP_ALLOCATED	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
SORT_OVERFLOW_PERCENT	The percentage, within the total number of sorts, of sorts that ran out of sort heap and may have required disk space for temporary storage. If the percentage is high, consider adjusting the database configuration by increasing the sort heap.
SORT_OVERFLOWS	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage. When a sort overflows, additional overhead is incurred because the sort requires a merge phase and might require more I/O if data needs to be written to disk.
SORT_SHRHEAP_ALLOCATED	The total amount of shared sort memory allocated in the database. If this value is often much higher or lower than the current shared sort memory threshold, the threshold should probably be adjusted. NOTE: The "shared sort memory threshold" is determined by the value of the SHEAPTHRES database manager configuration parameter if the SHEAPTHRES_SHR database configuration parameter is 0. Otherwise, it is determined by the value of SHEAPTHRES_SHR.
SORT_SHRHEAP_TOP	The database-wide shared sort memory high-water mark (in 4k pages).
STATIC_SQL_STMTS	The number of static SQL statements attempted.
STATIC_SQL_STMTS_RATE	The number, per second, of static SQL statements that were run during the specified time range.

Metric	Description
TOTAL_SORT_TIME_S	The total elapsed time (in seconds) for all sorts that were run. This metric can be used with TOTAL_SORTS to calculate the average sort time, which can indicate whether sorting is a performance issue.
TOTAL_SORTS	The total number of sorts that have been run. This metric can be used with SORT_OVERFLOWS to calculate the percentage of sorts that need more heap space. You can also use it with TOTAL_SORT_TIME to calculate the average sort time. If the amount of sort overflows is small relative to the total sorts, then increasing the sort heap size may have little impact on performance, unless it is increased substantially.
TOTAL_SUBAGENT_CONS	The number of connections made by a subagent to the database at the node.
XACT	The total number of units of work, calculated by taking the sum of INT_COMMIT, INT_ROLLBACK, COMMIT_SQL_STMTS and ROLLBACK_SQL_STMTS.
XACT_RATE	The number, per second, of transactions that were carried out during the specified time range.

HADR Activity Collection

The following table provides a list of the HADR Activity metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
HADR_ROLE	The current HADR role of the database. The data type of this element is integer. The value for this element is one of the following constants: SQLM_HADR_ROLE_STANDARD (value = 0): The database is not a HADR database. SQLM_HADR_ROLE_PRIMARY (value = 1): The database is the primary HADR database. SQLM_HADR_ROLE_STANDBY (value = 2): The database is the standby HADR database.
HADR_STATE	The current HADR state of the database. The data type of this element is integer. This element should be ignored if the HADR role of the database is standard. If the database has a HADR role of primary or standby, the value for this element is one of the following constants: SQLM_HADR_STATE_DISCONNECTED (value = 0): The database is not connected to its partner database. SQLM_HADR_STATE_LOC_CATCHUP (value = 1): The database is doing local catch-up. SQLM_HADR_STATE_REM_CATCH_PEND (value = 2): The database is waiting to connect to its partner to do remote catch-up. SQLM_HADR_STATE_REM_CATCHUP (value = 3): The database is doing remote catch-up. SQLM_HADR_STATE_PEER (value = 4): The primary and standby databases are connected and are in peer state.

Metric	Description
HADR_SYNCMODE	The current HADR synchronization mode of the database. The data type of this element is integer. This element should be ignored if the HADR role of the database is standard. If the database has a HADR role of primary or standby, the value for this element is one of the following constants: SQLM_HADR_SYNCMODE_SYNC: Sync mode SQLM_HADR_SYNCMODE_NEARSYNC: Nearsync mode SQLM_HADR_SYNCMODE_ASYNC: Async mode
HADR_CONNECT_STATUS	This element should be ignored if the HADR role of the database is standard. If the database has a HADR role of primary or standby, the value for this element is one of the following constants: SQLM_HADR_CONN_CONNECTED (value = 0): The database is connected to its partner node. SQLM_HADR_CONN_DISCONNECTED (value = 1): The database is not connected to its partner node. SQLM_HADR_CONN_CONGESTED (value = 2): The database is connected to its partner node, but the connection is congested. A connection is congested when the TCP/IP socket connection between the primary and standby pair is still alive, but one end cannot send to the other end. For example, the receiving end is not receiving from the socket connection, resulting in a full TCP/IP send space. The reasons for network connection congestion include the following: • The network is being shared by too many resources or the network is not fast enough for the transaction volume of the primary HADR node. • The server on which the standby HADR node resides is not powerful enough to retrieve information from the communication subsystem at the necessary rate.
HADR_CONNECT_TIME	Shows one of the following: HADR connection time HADR congestion time HADR disconnection time This element should be ignored if the HADR role of the database is standard. If the database HADR role is primary or standby, the meaning of this element depends on the value of the HADR_CONNECT_STATUS element: If the value of the HADR_CONNECT_STATUS element is SQLM_HADR_CONN_CONNECTED, then this element shows connection time. If the value of the HADR_CONNECT_STATUS element is SQLM_HADR_CONN_CONGESTED, then this element shows the time when congestion began. If the value of the HADR_CONNECT_STATUS element is SQLM_HADR_CONN_DISCONNECTED, then this element shows disconnection time. If there has been no connection since the HADR engine dispatchable unit (EDU) started, connection status is reported as Disconnected and HADR EDU startup time is used for the disconnection time. Since HADR connect and disconnect events are relatively infrequent, the time is collected and reported even if the DFT_MON_TIMESTAMP switch is off.

Metric	Description
HADR_HEARTBEAT	The number of missed heartbeats on the HADR connection. If the HADR role of the database is primary or standby, this element indicates the health of the HADR connection. A heartbeat is a message sent from the other HADR database at regular intervals. If the value for this element is zero, no heartbeats have been missed and the connection is healthy. The higher the value, the worse the condition of the connection. A HADR database expects at least one heartbeat message from the other database in each quarter of the time interval defined in the HADR_TIMEOUT database configuration parameter or in 30 seconds, whichever is shorter. For example, if the HADR_TIMEOUT value is 80 (seconds), then the HADR database expects at least one heartbeat message from the other database every 20 seconds. **NOTE:** The data type of this element is integer. **NOTE:** This element should be ignored if the HADR role of the database is standard. Use this element to determine the health of the HADR connection. Use the HADR_ROLE monitor element to determine the HADR role of the database.
HADR_LOCAL_HOST	The local HADR host name. The value is displayed as a host name string or an IP address string such as 1.2.3.4. This element should be ignored if the HADR role of the database is standard.
HADR_LOCAL_SERVICE	The local HADR TCP service. This value is displayed as a service name string or a port number string. This element should be ignored if the HADR role of the database is standard.
HADR_REMOTE_HOST	The remote HADR host name. The value is displayed as a host name string or an IP address string such as 1.2.3.4. This element should be ignored if the HADR role of the database is standard.
HADR_REMOTE_SERVICE	The remote HADR TCP service. This value is displayed as a service name string or a port number string. This element should be ignored if the HADR role of the database is standard.
HADR_REMOTE_INSTANCE	The remote HADR instance name. This element should be ignored if the HADR role of the database is standard.
HADR_TIMEOUT	The number of seconds it takes for a HADR database server to consider a communication attempt to have failed. For an attempt to fail, a HADR database server must not receive a reply message from its partner within the number of seconds specified in this element. This element should be ignored if the HADR role of the database is standard.
HADR_PRIMARY_LOG_FILE	The name of the current log file on the primary HADR database. This element should be ignored if the HADR role of the database is standard.
HADR_PRIMARY_LOG_PAGE	The page number in the current log file indicating the current log position on the primary HADR database. The page number is relative to the log file. For example, page zero is the start of the file. This element should be ignored if the HADR role of the database is standard.

Metric	Description
HADR_PRIMARY_LOG_LSN	The current log position of the primary HADR database. Log sequence number (LSN) is a byte offset in the log stream of the database. This element should be ignored if the HADR role of the database is standard.
HADR_STANDBY_LOG_FILE	The name of the current log file on the standby HADR database. This element should be ignored if the HADR role of the database is standard.
HADR_STANDBY_LOG_PAGE	The page number in the current log file indicating the current log position on the standby HADR database. The page number is relative to the log file. For example, page zero is the start of the file. This element should be ignored if the HADR role of the database is standard.
HADR_STANDBY_LOG_LSN	The current log position of the standby HADR database. Log sequence number (LSN) is a byte offset in the log stream of the database. This element should be ignored if the HADR role of the database is standard.
HADR_LOG_GAP_KB	The size (in KB) of the gap between the primary LSN and the standby LSN. When a log file is truncated, the LSN in the next log file starts as if log file truncation took place, as a result of that creating an LSN hole that contains no log data. Such holes can lead to a situation where the log gap does not reflect the actual log difference between the primary and standby LSN.
HADR_LOG_GAP_KB_RATE	The running log gap rate (in KB/s) between the primary and the standby database.

Host Properties Collection

The following table provides a list of the Host Properties metrics that are collected, and a description for each.

Metric	Description
• DATABASE	The name of the monitored database.
• PARTITION#	The partition number.
• PHYSICAL_HOST_NAME	The physical name of the host.
• MACHINE_MODEL	The machine model. Example: Sun Fire V240
• OPERATING_SYSTEM	The operating system. Example for OS and OS version collections: 64-bit sparcv9 sun4u
• OPERATING_SYSTEM_VERSION	The version of the operating system.
• RAM	The amount of machine RAM configured for DB2 servers.
• NUMBER_OF_CPU	The total number of CPU units available for DB2 servers.

Metric	Description
• NUMBER_OF_IO_DEVICES	The number of IO devices configured on the host.
• NUMBER_OF_NETWORK_INTERF ACES	The number of network interfaces installed on the host.

OS General Collection

The following table provides a list of the OS General metrics that are collected, and a description for each.

Metric	Description
DATABASE	The name of the monitored database.
DISK_UTILIZATION	The percentage of time the busiest device spent serving system-wide I/O requests. This metric serves as a measure for the system I/O load.
FREE_RAM	The amount of RAM that is free and not used by any of the running processes on the DB2 host.
PARTITION#	The partition number.
RESIDENT_SET_SIZE	The host OS resident memory set size.
SYSTEM_CPU_UTILIZATION	The total CPU consumption by all system processes (including DB2 processes) on the host.
SYSTEM_KERNAL_CPU_UTILIZA TION	The percentage of CPU time consumed by the operating system's kernel mode activities (processes).
SYSTEM_USER_CPU_UTILIZATI ON	The percentage of CPU time consumed by the operating system's user mode activities (processes).
USED_RAM	The total amount of memory (both resident and swapped RAM) consumed by all the operating system's processes, including DB2 (measured in MB).
USED_RAM_PCT	The percentage of RAM that is used by all the operating system's processes.
VIRTUAL_SET_SIZE	The host OS virtual memory set size.

Instance Activity Collection

The following table provides a list of the Instance Activity metrics that are collected, and a description for each.

Metric	Description
AGENTS_CREATED_EMPTY_P OOL	The number of agents created due to an empty agent pool, including the number of agents started at DB2 start-up (num_initagents).
AGENTS_REGISTERED	The number of agents registered in the currently monitored database manager instance (coordinator agents and subagents).
AGENTS_STOLEN	Represents the number of idle agents associated with an application that get reassigned to work on a different application.
CON_LOCAL_DBASES	The number of local databases that have applications connected.

Metric	Description
DB2_STATUS	This interface returns a text identifier, which is based on the definitions in the *sqlmon.h* file and can have one of the following values: ACTIVE QUIESCE_PEND QUIESCED
DB2START_TIME	The date and time that the database manager was started using the db2start command.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.
LAST_RESET	The date and time that the monitor counters were reset.
LOCAL_CONNECTIONS	The number of local applications that are currently connected to a database within the database manager instance being monitored.
LOCAL_CONN_EXEC	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
NUM_NODES_IN_DB2_INSTANCE	The number of partitions on the instance.
PERCENT_OF_AGENTS_IN_USE	Of the maximum agents, the percentage of those agents in use.
PIPED_SORTS_ACCEPTED	The number of piped sorts that have been accepted. Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. When the number of accepted piped sorts is low compared to the number requested, you can improve sort performance by adjusting one or both of the following configuration parameters: sortheap, sheapthres
PIPED_SORTS_REQUESTED	The number of piped sorts that have been requested.
POST_THRESHOLD_SORTS	The number of sorts that have requested heaps after the sort heap threshold has been exceeded. If this element's value is high, you can increase the sort heap threshold (sheapthres) or adjust applications to use fewer or smaller sorts through SQL query changes.
PRODUCT_NAME	The details of the version of the DB2 instance that is running.
REMOTE_CONN_EXEC	The number of remote applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
REMOTE_CONNECTIONS	The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.
SERVICE_LEVEL	The current corrective service level of the DB2 instance.
SORT_HEAP_ALLOCATED	The total number of allocated pages of sort heap space for all sorts.

Instance General Collection

The following table provides a list of the Instance General metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
IS_PARTITIONABLE	Indicates whether the instance is partitioned. Values: 1 — Partitioned 0 — Not partitioned **Note:** An instance with one partition is considered not partitioned.
IS_PURESCALE	Indicates whether the instance is pureScale. Values: 1 — pureScale 0 — Not pureScale
CONFIGURATION	Indicates the configuration of the instance. Values: Partitioned Non-Partitioned Cluster (for pureScale)
NUM_NODES	The number of members in the instance (partitions in a DPF, and members in a pureScale environment).

Instance Nodes Collection

The following table provides a list of the Instance Nodes (partitions or members) metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
NODE_ID	The ID of the partition or member.
HOST_NAME	The machine that was associated with the member when it was first added to the instance.
CURRENT_HOST_NAME	The host on which the node is running. In a pureScale environment, this could be a different host than the one originally configured.
MEMBER_TYPE	Indicates the member type. Values: MEMBER CF
DB_PARTITION_NUM	The partition number. For a pureScale instance, this is always 0.

Metric	Description
STATE	The state of the member or cluster. The states for a member are: STARTED STOPPED RESTARTING, WAITING_FOR_FAILBACK ERROR UNKNOWN The states for a cluster caching facility are: STOPPED RESTARTING BECOMING_PRIMARY PRIMARY CATCHUP1 PEER ERROR UNKNOWN.
ALERT	Provides information about alerts on the instance. Values: YES — There is at least one alert. NO — There are no alerts. NULL — Inapplicable

Instance Usability Collection

The following table provides a list of the Instance Usability metrics that are collected, and a description for each.

Metric	Description
INSTANCE	The monitored instance.
INSTANCE_AVAILABILITY	The percentage of instance partitions available for the currently running query.

IO Activity Collection

The following table provides a list of the IO Activity metrics that are collected, and a description for each.

Metric	Description
ASYNC_PAGES_READ_PER_REQUEST	The average number of pages in each asynchronous read request.
ASYNC_READ	The total number of asynchronous reads (pages) during the specified time range.
ASYNC_READ_RATE	The number of asynchronous reads (pages) per second during the specified time range.
ASYNC_READ_REQUESTS	The total number of asynchronous read requests during the specified time range.

Metric	Description
ASYNC_WRITES	The total number of asynchronous writes (pages) during the specified time range.
ASYNC_WRITES_RATE	The number of asynchronous writes (pages) per second during the specified time range.
BALANCE_AVERAGE_LOGICAL_READ D	The balance of logical reads across the partitions. This metric shows how balanced the DB2 is, across its partitions, in terms of the logical reads operations.
BALANCE_AVERAGE_PHYSICAL_READ D	The balance of physical reads across the partitions. This metric shows how balanced the DB2 is, across its partitions, in terms of the physical reads operations.
DATA_HIT_RATIO	The hit ratio of data reads that were satisfied from the buffer pool, without having to perform physical read operations.
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
DIRECT_READ_REQS	The number of requests to perform a direct read of one or more sectors of data. To calculate the average number of sectors read by a direct read, you can use the following formula: DIRECT_READS / DIRECT_READ_REQS
DIRECT_READ_TIME	The elapsed time (in milliseconds) required to perform the direct reads.
DIRECT_READS	The number of read operations that do not use the buffer pool. To calculate the average number of sectors read by a direct read, you can use the following formula: DIRECT_READS / DIRECT_READ_REQS
DIRECT_READS_RATE	The direct reads rate (pages per second).
DIRECT_WRITE_REQS	The number of requests to perform a direct write of one or more sectors of data. To calculate the average number of sectors written by a direct write, you can use the following formula: DIRECT_WRITES / DIRECT_WRITE_REQS
DIRECT_WRITE_TIME	The elapsed time (in milliseconds) required to perform the direct writes.
DIRECT_WRITES	The number of write operations that do not use the buffer pool. To calculate the average number of sectors that written by a direct write, use the following formula: DIRECT_WRITES / DIRECT_WRITE_REQS
DIRECT_WRITES_RATE	The direct writes rate (pages per second).

Metric	Description
FILES_CLOSED	The total number of database files closed.
	The database manager opens files for reading and writing into and out of the buffer pool. The maxfilop configuration parameter controls the maximum number of database files open by an application at any time. If that maximum is about to be reached, a file will be closed before a new one is opened.
	NOTE: The number of files opened may not equal the number of files closed.
INDEX_HIT_RATIO	The hit ratio of index reads that were satisfied from the buffer pool, without having to perform index physical read operations.
LOG_CLEANS_RATIO	The percentage of LSN gap cleans within the total number of cleans.
LOGICAL_READS	The total number of pages that have been requested from the buffer pool (logical),
LOGICAL_READS_RATE	The number of logical reads (pages) per second during the specified time range.
OVERALL_POOL_HIT_RATIO	The overall hit ratio of logical reads to physical reads. That is, the percentage of database page I/O requests that were satisfied from the buffer cache and, therefore, did not have to perform disk reads. This ratio measures how efficiently DB2 is using the memory allocated to its buffer cache.
PHYSICAL_READS	The number of physical reads (pages).
	Physical reads are used when a connection requests a page that is not already in the buffer cache.
	Physical read operations from the disk are necessary when accessing new table or index pages. Nevertheless, these operations should be avoided when possible, as requested pages should reside in the DB2 buffer cache memory.
PHYSICAL_READS_RATE	The number of physical reads (pages) per second during the specified time range.
PHYSICAL_WRITES	The total number of times a buffer pool page was physically written to disk.
PHYSICAL_WRITES_RATE	The number of buffer pool pages physically written to disk per second.
POOL_ASYNC_DATA_READ_REQS	The number of asynchronous read requests made by the prefetcher to the operating system. These are typically large block IOs of multiple pages.
POOL_ASYNC_DATA_READS	The number of data pages read in from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces.
POOL_ASYNC_DATA_WRITES	The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.
POOL_ASYNC_INDEX_READ_REQS	The number of asynchronous read requests for index pages.

Metric	Description
POOL_ASYNC_INDEX_READS	The number of index pages read in from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces.
POOL_ASYNC_INDEX_WRITES	The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.
POOL_ASYNC_READ_TIME	The total amount of time spent reading in data and index pages from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces. The value is given in milliseconds.
POOL_ASYNC_WRITE_TIME	The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
POOL_ASYNC_XDA_READ_REQS	The number of asynchronous read requests for XML storage object (XDA) data.
POOL_ASYNC_XDA_READS	The number of XML storage object (XDA) data pages read in from the tablespace containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of tablespaces.
POOL_ASYNC_XDA_WRITES	The number of times a buffer pool data page for an XML storage object (XDA) was physically written to disk by either an asynchronous page cleaner or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.
POOL_DATA_L_READS	The number of data pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_DATA_P_READS	The number of data pages read in from the tablespace containers (physical) for regular and large tablespaces.
POOL_DATA_WRITES	The number of times a buffer pool data page was physically written to disk.
POOL_DRTY_PG_STEAL_CLNS	The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
POOL_DRTY_PG_THRSH_CLNS	The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.
POOL_INDEX_L_READS	The number of index pages that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_INDEX_P_READS	The number of index pages read in from the tablespace containers (physical) for regular and large tablespaces.
POOL_INDEX_WRITES	The number of times a buffer pool index page was physically written to disk.
POOL_LSN_GAP_CLNS	The number of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database. This element can be used to help evaluate whether you have enough space for logging or whether you need more log files or larger log files.

Metric	Description
POOL_NO_VICTIM_BUFFER	The number of times an agent did not have a preselected victim buffer available. This element can be used to help evaluate whether you have enough page cleaners for a given buffer pool when using proactive page cleaning.
POOL_READ_TIME	The total amount of time spent reading in data and index pages from the tablespace containers (physical) for all types of tablespaces. The value is given in milliseconds.
POOL_TEMP_DATA_L_READS	The number of data pages that have been requested from the buffer pool (logical) for temporary tablespaces.
POOL_TEMP_DATA_P_READS	The number of data pages read in from the tablespace containers (physical) for temporary tablespaces.
POOL_TEMP_INDEX_L_READS	The number of index pages that have been requested from the buffer pool (logical) for temporary tablespaces.
POOL_TEMP_INDEX_P_READS	The number of index pages read in from the tablespace containers (physical) for temporary tablespaces.
POOL_TEMP_XDA_L_READS	The number of pages for XML storage object (XDA) data that have been requested from the buffer pool (logical) for temporary tablespaces.
POOL_TEMP_XDA_P_READS	The number of pages for XML storage object (XDA) data read in from the tablespace containers (physical) for temporary tablespaces.
POOL_WRITE_TIME	The total amount of time spent physically writing data or index pages from the buffer pool to disk. The value is given in milliseconds.
POOL_XDA_L_READS	The number of data pages for XML storage objects (XDAs) that have been requested from the buffer pool (logical) for regular and large tablespaces.
POOL_XDA_P_READS	The number of data pages for XML storage objects (XDAs) read in from the tablespace containers (physical) for regular and large tablespaces.
POOL_XDA_WRITES	The number of times a buffer pool data page for an XML storage object (XDA) was physically written to disk.
PREFETCH_WAIT_TIME	The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool.
SYNC_READS	The total number of synchronous reads (pages) during the specified time range.
SYNC_READS_RATE	The number of synchronous reads (pages) per second during the specified time range.
SYNC_WRITES	The total number of synchronous writes (pages) during the specified time range.
SYNC_WRITES_RATE	The number of synchronous writes (pages) per second during the specified time range.
THRESHOLD_CLEANS_RATIO	The percentage of all page cleans that took place after a buffer pool had reached the criterion defined within the database for dirty page threshold.

Metric	Description
UNREAD_PREFETCH_PAGES	The number of pages that the prefetcher read in that were never used.
	If this number is high, prefetchers are causing unnecessary I/O by reading pages into the buffer pool that are not being used.
VICTIM_CLEANS_RATIO	The percentage of all page cleans that took place because a synchronous write was needed during the victim buffer replacement for the database.
	A victim page is a clean or dirty page in the buffer pool that is removed because DB2 needs to make room for incoming pages. If a victim page is a dirty page, then the information must be written out to disk.
XML_HIT_RATIO	The hit ratio of XML reads that were satisfied from the buffer pool, without having to perform physical read operations.

License Details Collection

The following table provides a list of the License Details metrics that are collected, and a description for each.

Metric	Description
IS_LICENSED	Indicates whether the installed product is licensed (1) or not (0).
INSTANCE_NAME	The name of the monitored instance.
PROD_RELEASE	The product release number.
INSTALLED_PROD	Represents the DB2 products that are installed on the system.

Log Activity Collection

The following table provides a list of the Log Activity metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
TOTAL_LOG_AVAILABLE_KB	Use this element with TOTAL_LOG_USED_KB to determine whether you need to adjust the following configuration parameters to avoid running out of log space:
	logfilsiz
	logprimary
	logsecond
TOTAL_LOG_USED_KB	The total amount of active log space currently used (in KB) in the database.
LOG_UTILIZATION	The percentage of used log within the entire available log.
SECONDARY_LOG_USED_TOP_KB	The maximum amount of secondary log space used (in KB).
	The value is zero if the database does not have any secondary log files (which is the case if there are none defined).

Metric	Description
TOTAL_LOG_USED_TOP_KB	The maximum amount of total log space used (in KB).
	You can use this element to help evaluate the amount of primary log space that you have allocated. Comparing the value of this element with the amount of primary log space you have allocated can help you to evaluate your configuration parameter settings. Your primary log space allocation can be calculated using the following formula:
	logprimary * logfilsiz * 4096
	You can use this element with SECONDARY_LOG_USED_TOP_KB and SECONDARY_LOGS_ALLOCATED to evaluate your current dependency on secondary logs.
SECONDARY_LOGS_ALLOCATED	You may use this element with SEC_LOG_USED_TOP and TOT_LOG_USED_TOP to show your current dependency on secondary logs. If this value is consistently high, you may need larger log files, more primary log files, or more frequent COMMIT statements within your application.
	As a result, you may need to adjust the following configuration parameters:
	logfilsiz
	logprimary
	logsecond
	logretain
LOG_READS	The number of log reads (pages) during the specified time range.
LOG_READS_RATE	The number of log reads (pages) per second during the specified time range.
LOG_READ_TIME_S	The total elapsed time spent by the logger reading log data from the disk.
LOG_WRITES	The number of log writes (pages) during the specified time range.
LOG_WRITES_RATE	The number of log writes (pages) per second during the specified time range.
LOG_WRITE_TIME_S	The total elapsed time spent by the logger writing log data to the disk.
NUMBER_LOG_WRITE_IO	The number of I/O requests issued by the logger for writing log data to the disk.
NUMBER_LOG_READ_IO	The number of I/O requests issued by the logger for reading log data from the disk.
NUMBER_LOG_PART_PAGE_IO	The number of I/O requests issued by the logger for writing partial log data to the disk.
NUMBER_LOG_BUFFER_FULL	The number of times agents that copy log records into the log buffer have to wait for log data to write to disk. This value is incremented each time an agent has to wait. For example, if two agents are waiting, the value is incremented by two.
NUMBER_LOG_DATA_FOUND_IN_BUFFER	The number of times an agent reads log data from the buffer. Reading log data from the buffer is preferable to reading from the disk because the latter is slower.
	Use this element with the NUM_LOG_READ_IO element to determine if the LOGBUFSZ database configuration parameter needs to be increased.

Metric	Description
APPL_ID_OLDEST_XACT	The application ID (which corresponds to the AGENT_ID value from the application snapshot) of the application that has the oldest transaction.
	This element can help in determining which application has the oldest active transaction. That application can be forced to free up log space. If it is taking up a great deal of log space, examine the application to determine if it can be modified to commit more frequently.
	There are times when there is not a transaction holding up logging, or the oldest transaction does not have an application ID (for example, an in-doubt transaction or an inactive transaction). In these cases, an application ID is not returned in the data stream.
LOG_TO_REDO_FOR_RECOVERY _KB	The amount of log (in KB) that has to be redone for crash recovery.
	When the snapshot is taken, this value is calculated based on the conditions at the time of the snapshot. Larger values indicate longer recovery times after a system crash. If the value seems excessive, check the LOG_HELD_BY_DIRTY_PAGES element to see if page cleaning needs to be tuned. Also, check if there are any long running transactions that need to be terminated.
LOG_HELD_BY_DIRTY_PAGES_K B	The amount of log corresponding to the difference between the oldest dirty page in the database and the top of the active log.
	When the snapshot is taken, this value is calculated based on the conditions at the time of the snapshot.
	Use this element to evaluate the effectiveness of page cleaning for older pages in the buffer pool.
	The cleaning of old pages in the buffer pool is governed by the softmax database configuration parameter. If the page cleaning is effective, then LOG_HELD_BY_DIRTY_PAGES_KB should be less than or approximately equal to:
	(softmax / 100) * logfilsiz * 4096. If this statement is not true, increase the number of page cleaners using the num_iocleaners configuration parameter.
	If the condition is true and it is desired that less log be held by dirty pages, then decrease the softmax configuration parameter.

Log File Collection

The following table provides a list of the Log File metrics that are collected.

Metric	Description
INSTANCE	The name of the monitored instance.
FULL_DIAG_LOG_PATH	The full path and name of the diag log file (the filename is db2diag.log).
DIAG_LOG_SIZE	The diag log file size.

Log Message Collection

The following table provides a list of the Log Message metrics that are collected, and a description for each.

Metric	Description
APPL_ID	The application ID that is related to the message in the notification log file.
DBNAME	The name of the monitored database.
DBPARTITIONNUM	The name of the monitored database.
FUNCTION	The name of the function that generated the message. This metric is retrieved for version 10 and higher.
IMPACT	Qualifies the impact of this message from a user's perspective. This clarifies the impact that the event described by the message has on the business process DB2 is part of. The impact is determined according to DB2 definition. The possible values are: • CRITICAL • IMMEDIATE • NONE • POTENTIAL • UNLIKELY This metric is reserved for version 10 and higher.
MSG	The message text.
MSGNUM	The message number.
MSGSEVERITY	The message severity. Values: C (critical) E (error) W (warning) I (informational) NULL (if the message severity could not be determined)
MSGTYPE	The message type.
PID	The process ID that caused this message.
PROCESSNAME	The process name.
TID	The thread ID.
TIMESTAMP	The log message timestamp.

Messages Count Collection

The following table provides a list of the Messages Count metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
CRITICAL_COUNT	The number of critical messages for the selected period.
FATAL_COUNT	The number of fatal messages for the selected period.
WARNING_COUNT	The number of warning messages for the selected period.

Metric	Description
INFORMATIONAL_COUNT	The number of informational messages for the selected period.
OTHER_COUNT	The number of unknown messages for the selected period.

Monitor Switches Collection

The following table provides a list of the Monitor Switches metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
UOW_SW_STATE	The state (0 or 1) of the unit of work monitor recording switch.
STATEMENT_SW_STATE	The state (0 or 1) of the STATEMENT recording switch.
LOCK_SW_STATE	The state (0 or 1) of the LOCK recording switch.
SORT_SW_STATE	The state (0 or 1) of the SORT recording switch.
TABLE_SW_STATE	The state (0 or 1) of the TABLE recording switch.
BUFFERPOOL_SW_STATE	The state (0 or 1) of the BUFFERPOOL recording switch.
TIMESTAMP_SW_STATE	The state (0 or 1) of the TIMESTAMP recording switch.

OS DB2 General Process Collection

The following table provides a list of the OS DB2 General Process metrics that are collected, and a description for each.

Metric	Description
DATABASE	The name of the monitored database.
PARTITION#	The partition number.
ALL_AGENTS_MEMORY_USAGE_INSTANCE	The entire agents process memory consumption, both RAM resident and swapped.
ALL_AGENTS_MEMORY_USAGE_DATABASE	The entire agents process memory consumption, both RAM resident and swapped.
ALL_AGENTS_CPU_UTILIZATION	The overall CPU utilization (%) of the DB2 agents.
NON_DB2_CPU_UTILIZATION	The overall CPU consumption not associated with the monitored DB2 database partition (background and foreground processes).
BALANCE_DB2_CPU_USAGE	The balance of CPU usage between all monitored partitions.
BALANCE_DB2_MEMORY_USAGE	The balance of memory usage (%) between all monitored partitions.

Metric	Description
ALL_AGENTS_MEMORY_USAGE_TOTAL	The overall monitored DB2 MEMORY usage (MB).
ALL_AGENTS_MEMORY_USAGE_UTILIZATION	The overall monitored DB2 MEMORY utilization (%), which is the amount of memory used by DB2, as a percentage of the DB2 host's configured memory.

Partition Tablespaces Summary Collection

The following table provides a list of the Partition Tablespaces Summary metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The partition number.
OVERALL_FIXED_TABLESPACE_TOTAL_SIZE_MB	The total size (in MB) of the fixed DMS tablespaces.
OVERALL_FIXED_TABLESPACE_USED_SIZE_MB	The overall fixed DMS tablespace size (in MB) used.
OVERALL_FIXED_TABLESPACE_UTILIZATION	The overall utilization (%) of the disk space that is allocated for use by DB2 fixed-size tablespaces.
OVERALL_AUTO_TABLESPACE_USED_SIZE_MB	The overall total size (in MB) of the tablespaces used.
OVERALL_AUTO_TABLESPACE_UTILIZATION	The overall utilization (%) of the file system space that is available for DB2, for carrying out auto storage tablespace operations.
OVERALL_TOTAL_SIZE_MB	The total allocated size (in MB) of all the tablespaces in the partition.
OVERALL_USED_SIZE_MB	The overall used size (in MB) of all the tablespaces.
BALANCE_OVERALL_USED_SIZE_MB	The balance of storage used across the partitions. This metric shows the level of balance the DB2 has, in terms of storage capacity, across its partitions.
OVERALL_TOTAL_USED_SPACE_DELTA	The space used by the delta of all the tablespaces.
OVERALL_TOTAL_USED_SPACE_DELTA_RATE	The used space delta per second.
AUTO_STORAGE_STATE	The auto storage state. Values: 0 — Fixed 1 — Auto 0,1 — Both

Partition Usability Collection

The following table provides a list of the Partition Usability metrics that are collected, and a description for each.

Metric	Description
INSTANCE	The name of the monitored instance.
PARTITION_NUMBER	The partition number.
PARTITION_HOST_NAME	The partition host name.
PARTITION_AVAILABILITY	The availability of the partition, which reflects the instance usability.
ERROR_MSG_OBS	Indicates the error message received while attempting to connect to the database.
OS_CONNECT_AVAILABILITY	Indicates the OS connect availability with the associated OS credential. Values: 0 — Connection to the remote host failed. 100 — Connection to the remote host succeeded.
OS_CONNECTION_TIME	The average duration (in milliseconds) for performing a single connection to the database.

pureScale Alerts Collection

Describes the metrics collected about instances hosted in a DB2 pureScale environment.

Metric	Description
INSTANCE_NAME	The name of the instance hosted in a pureScale environment.
PURESCALE_ALERTS_NO	Total number of alerts DB2 reported on the pureScale environment.
MESSAGE	Information about the alert.
ACTION	The action required to clear the alert.
IMPACT	The impact to the DB2 pureScale instance if the alert is not cleared.

Top SQLs Collection

The following table provides a list of the Top SQLs metrics that are collected, and a description for each.

Metric	Description
DB_NAME	Name of the Monitored database
MEMBER	The database partition/member from which the data was retrieved for this row.
EXECUTABLE_ID	An opaque binary token generated on the data server that uniquely identifies the SQL statement section that was executed.
SECTION_TYPE	Type of SQL. Can be 'S' for static SQL or 'D' for dynamic. Static SQLs are only available in DB2 version 9.7.0.1 and above.

Metric	Description
STMT_TYPE_ID	Statement type identifier. Possible values are: Statement not prepared DDL, (not Set Constraints) DDL, Set Constraints DML, Select DML, Insert/Update/Delete Authorization DML, Select (blockable) DML, Lock Table DML, Commit/Rollback Set environment DDL, Savepoint DDL, (declared user temp) Passthru support CALL Free locator DML, Select with IUD DML, Select with IUD (blockable) Top-level SET, no SQL Top-level SET, reads SQL DDL, (issues internal commit) Top-level SET, modifies SQL Unknown Parent topic: Database system monitor elements
HASH_VALUE	Identifier of the SQL by hash value.
NET_ELAPSED_TIME	The total time in seconds that was spent executing the particular statement, excluding time spent executing routines used as part of the statement.
NUM_EXECUTIONS	The number of times that an SQL statement has been executed. Can be used to identify the most frequently executed SQL statements in your system.
NUM_EXECUTIONS_WITH_METRICS	Number of executions that include performance metrics. Used to calculate the averages correctly.
ROWS_READ	The number of rows read from the table. This element helps you identify tables with heavy usage for which you may want to create additional indexes.
ROWS_RETURNED	The number of rows that have been selected and returned to the application. If this number is significantly lower then "Rows read", then it might indicate an index is missing.
ROWS_MODIFIED	The number of rows changed (inserted, deleted or updated) by the statement. A high value for table-level information indicates there is heavy usage of the table and you may want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table.
QUERY_COST_ESTIMATE	Estimated cost for a query, as determined by the SQL compiler. This value is reported in timerons.

Metric	Description
TOTAL_ROUTINE_INVOCATIONS	The total number of times a routine was invoked by the statement.
DEADLOCKS	The number of times that a request to lock an object resulted in a deadlock.
LOCK_WAITS	The number of times that applications or connections waited for locks.
LOCK_TIMEOUTS	The number of times that a request to lock an object timed out instead of being granted.
STMT_SORTS	The total number of times that a set of data was sorted in order to process the statement operation. Can be used to help identify the need for an index, since indexes can reduce the need for sorting of data. Using the related elements in the above table you can identify the SQL statement for which this element is providing sort information, and then analyze this statement to determine index candidates by looking at columns that are being sorted (for example, columns used in ORDER BY and GROUP BY clauses and join columns).
SORT_OVERFLOWS	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
DIRECT_READS	The number of read operations that do not use the buffer pool.
DIRECT_WRITES	The number of write operations that do not use the buffer pool.
POOL_L_READS	Total logical Reads for the SQL, including data, index, temp, XML storage object (XDA) and temp_xda reads.
POOL_P_READS	Total Physical Reads for the SQL, including data, index, temp, XML storage object (XDA) and temp_xda reads
POOL_WRITES	The number of times a buffer pool page was physically written to disk. This includes both data pages as well as XML storage object (XDA) and index pages.
TOTAL_CPU_TIME_S	The total CPU time consumed by the statement.
LOCK_WAIT	The time spent waiting for locks.
COMPILE_TIME	The overall amount of time that was required to prepare the specific SQL statement.
CF_WAIT_TIME	The overall amount of time spent communicating with the cluster caching facility.
WLM_WAIT	The time spent waiting on a WLM queuing threshold.
DIRECT_READ_TIME	The elapsed time required to perform direct reads, i.e: reads that do not use the buffer pool.
DIRECT_WRITE_TIME	The elapsed time required to perform direct writes, i.e: reads that do not use the buffer pool.
POOL_READ_TIME	The elapsed time required to perform pool reads, i.e: reads that use the buffer pool.
POOL_WRITE_TIME	The elapsed time required to perform pool writes, i.e: reads that use the buffer pool.
PREFETCH_WAIT_TIME	The time spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool.

Metric	Description
RECLAIM_WAIT_TIME	In a DB2® pureScale® environment, this represents the amount of time spent waiting on page locks, where the lock request caused a page to be reclaimed.
SPACEMAPPAGE_RECLAIM_WAIT_TIME	In a DB2® pureScale® environment, this represents the amount of time spent waiting on page locks for pages related to internally maintained object space management where the lock request caused a reclaim from another member.
AUDIT_FILE_WRITE_WAIT_TIME	Time spent waiting to write an audit records.
AUDIT_SUBSYSTEM_WAIT_TIME	Time spent waiting for space in audit buffer. Waiting occurs when audit buffer is full and agent must wait for audit daemon to write buffer to disk.
DIAGLOG_WRITE_WAIT_TIME	The time spent waiting on a write to the db2diag log file.
EVMON_WAIT_TIME	The amount of time spent waiting for an event monitor record to become available.
IDA_RECV_WAIT_TIME	The time spent waiting to receive data from an in-database analytics process.
IDA_SEND_WAIT_TIME	The time spent waiting to send data to an in-database analytics process.
LOG_BUFFER_WAIT_TIME	The amount of time an agent spends waiting for space in the log buffer.
LOG_DISK_WAIT_TIME	The amount of time spent waiting for log records to be flushed to disk.
TOTAL_SECTION_PROC_TIME	The total amount of processing time spent performing section execution. Processing time does not include wait time.
TOTAL_SECTION_SORT_TIME	Total amount of time spent performing sorts while executing a section, which is the execution of the compiled query plan generated by the SQL statement.
FCM_RECV_WAIT_TIME	The time spent waiting for an FCM reply message containing the results of a previously sent FCM request message. This value reflects both the time required to send the response between partitions using FCM and the time required for the subagent to process the request message.
FCM_SEND_WAIT_TIME	The time spent blocking on an FCM message send. The value reflects the time spent blocking for FCM buffers to be flushed from an FCM channel when distributing internal requests on the database system.
TOTAL_ROUTINE_TIME	The total time spent executing routines.
TOTAL_SECTION_TIME	The total time agents spent performing section execution.
TOTAL_WAIT_TIME	Total time spent waiting within the DB2® database server, while processing an activity. These includes waits for things like I/O, Locks, Memory etc. The value is given in seconds.
WAIT_FOR_CPU_S	The total time that requests, that were run in a service class, spent waiting to access the CPU.
TOTAL_EXTENDED_LATCH_WAIT_TIME	The amount of time, spent in extended latch waits.

Metric	Description
STMT_ELAPSED_TIME	The total time in seconds that was spent executing the particular statement including all routines that were called.
TOTAL_HIT_RATIO	A ratio indicating the amount of physical reads as oppose to logical overall reads. The indicator is calculated using the formula: 100 * (1 - (POOL_P_READS /POOL_L_READS).
IO_WAIT	Total time spent waiting, while processing an I/O request. Include read and write operations, both direct and from memory pools.
EXECUTE_TIME	Total time spent executing routines and performing section execution. Processing time does not include wait time.
LOG_WAIT	Total time spent waiting for log records to be flushed to disk or waiting for space in the log buffer.
OTHER_WAIT	Total time spent waiting for miscellaneous events such as audit records writes, dialog writes, event monitor records and send or receive data from an in-database analytics process.
FCM_WAIT	The time spent waiting for an FCM message to be sent or FCM reply to be received.
PURESCALE_WAIT	In a DB2® pureScale® environment, this is the time spent waiting for communicating with the cluster caching facility or waiting for page locks, where the lock request caused a page to be reclaimed either locally or from another member.
MEMORY_WAIT	The amount of time, spent in extended latch waits.
WAIT_TIME_PERCENT	Percent of time spent waiting within the DB2® database server out of the total time spent actively executing the statement.
OVERALL_SORT_TIME	Summary of the sort time of all the gathered SQL's.
OVERALL_CPU_WORKLOAD	Summary of the CPU of all the gathered SQL's.
TOTAL_NUM_EXECUTIONS	Summary of the SQL executions of all the gathered SQL's.
OVERALL_EXECUTION_TIME	Summary of the execution time of all the gathered SQL's.
OVERALL_POOL_L_READS	Summary of the data logical reads of all the gathered SQL's.
TOTAL_COMPILE_TIME	Summary of the prep time of all the gathered SQL's.
OVERALL_POOL_P_READS	Summary of the data physical reads of all the gathered SQL's.

Query Agents Details Collection

The following table provides a list of the Query Agents Details metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.

Metric	Description
FOREGROUND_BACKGROUND_TYPE	The foreground or background status of the application.
LAST_RESET	The date and time when the monitor counters were reset.
PACKAGE_CACHE_HIT_RATIO	A ratio that indicates how well the package cache is helping to avoid the reload of packages and sections for static SQL from the system catalogs and the recompilation of dynamic SQL statements. The indicator is calculated using the formula: (1-(PKG_CACHE_INSERTS/PGK_CACHE_LOOKUPS))*100
POOL_DATA_WRITES	The number of times a buffer pool data page was physically written to disk.
PERCENT_SORT_OVERFLOWS	The percentage of sort operations, out of the total sort operations, that ran out of sort heap and may have required disk space for temporary storage.

Database Partition Activity Collection

The following table provides a list of the Database Partition Activity metrics that are collected, and a description for each.

Metric	Description
DATABASE_NAME	The name of the monitored database
MEMBER	The member or partition ID.
TOTAL_RQSTS	The total amount of requests completed.
AVG_RESPONSE_TIME	The average duration (in seconds) for a single request.
TOTAL_RQST_TIME	The total amount of time (in seconds) spent working on requests.
BALANCE_TOTAL_RQST_TIME	The balance of the total amount of time spent working on requests across the database partitions. The balance is poor if one partition (or member) is doing most of the activity.
TOTAL_WAIT_TIME	The total amount of time (in seconds) spent waiting within the DB2 database server.
TOTAL_CPU_TIME	The total amount of CPU time used (in seconds) while within the DB2 database server. This includes both user and system CPU time.
BALANCE_TOTAL_CPU_TIME	The balance of CPU usage across all monitored partitions.
%WAIT	The amount of time the database spent waiting, as a percentage of the total request time.
TOTAL_CPU_WAIT	The total amount of time (in microseconds) that requests run in this service class spent waiting to access the CPU. This is only available for DB2 version 10 and higher.
FCM_REVC_VOLUME	The total amount of data (in bytes) received through the FCM communications layer.

Metric	Description
FCM_SEND_VOLUME	The total amount of data (in bytes) distributed by the FCM communications layer.
FCM_REVC_VOLUME_RATE	The total amount of data (in bytes) received through the FCM communications layer.
FCM_SEND_VOLUME_RATE	The total amount of data (in bytes) distributed by the FCM communications layer.
TCPIP_SEND_VOLUME_KB	The amount of data (in KB) sent by the data server to the clients.
TCPIP_SEND_VOLUME_RATE	The number of KBs sent per second by the data server to the clients.
TCPIP_REVC_VOLUME_KB	The amount of data (in KB) received by the data server from the clients over TCP/IP.
TCPIP_REVC_VOLUME_RATE	The number of KBs received per second by the server from the clients.

DB2 Partition Subcategories Wait Collection

The following table provides a list of the DB2 Partition Subcategories Time-spent Monitor Element metrics that are collected, and a description for each.

Starting from DB2 v9.7.0.1, DB2 uses time-spent monitor elements to track how time is spent in the system. These elements show where time is spent waiting, or performing various types of processing. Time-spent monitor elements get rolled up into more general monitor elements. Elements with the same PARENT_METRIC_NAME are rolled up into that parent.

Metric	Description
DATABASE_NAME	The name of the monitored database
MEMBER	The database partition or member number.
METRIC_NAME	The event name to which the TOTAL_TIME metric value refers.
PARENT_METRIC_NAME	The identifier of the parent for the TOTAL_TIME metric, whose value contains the TOTAL_TIME value as a subset.
TOTAL_TIME	The total time value (in seconds), which corresponds to METRIC_NAME.
COUNT	The number of occurrences of this type of event.
COUNT_RATE	The rate of the number of occurrences of this type of event.

Metric	Description
TOTAL_TIME_RATE	The rate (per second) of the total time spent.
CATEGORY	The category of the event. Values:
	I/O wait - The amount of time spent waiting for IO. Subcategories provide additional information.
	Other wait - The aggregate of several events that do not typically have a high impact on performance. Subcategories provide additional information.
	WLM wait - The amount of time spent waiting for the WLM queue.
	pureScale wait - The amount of time spent waiting for various functions within the cluster. Subcategories provide additional information.
	Idle - The amount of time the client waited that was not spent within DB2 database server.
	Network wait - The amount of time spent waiting for network protocols. Subcategories provide additional information.
	Lock wait - The amount of time spent waiting for locks.
	Log wait - The amount of time spent waiting for the log. Subcategories provide additional information.
	Transaction - The amount of time spent on the commit or rollback of the transaction.
	Compile - The amount of time spent doing explicit or implicit compiles.
	Memory wait - The amount of time spent waiting for an internal latch.
	Administration - The amount of time spent processing administration tasks. Subcategories provide additional information.
	Execute - The amount of time spent executing the request. Subcategories provide additional information.
	FCM wait - The amount of time spent waiting for the FCM communications layer. This is only applicable on the partition level. Subcategories provide additional information.
	IPC - The amount of time spent waiting for IPC channels.

History Locks Wait Collection

The following table provides a list of the History Locks Wait metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
LOCK_NAME	The ID of the lock.

Metric	Description
AGENT_ID	A system-wide unique ID for the locked wait application. On a single-partitioned database, this ID consists of a 16-bit counter. On a multi-partitioned database, this ID consists of the coordinating partition number concatenated with a 16-bit counter. In addition, this ID will be the same on every partition where the application may make a secondary connection. The application handle (synonymous with agent ID) can be used to uniquely identify an active application. It can also be used as input to the FORCE APPLICATION command or API. On multi-node systems, this command can be issued from any node where the application has a connection. Its effect is global.
STMT_TEXT	The full statement text of the locked agent.
LOCK_MODE	The type of lock being held. This mode can assist in determining the source of the contention for resources. One of the following is indicated, depending on the type of monitoring information being examined: • The type of lock another application holds on the object that this application is waiting to lock (for application-monitoring and deadlock-monitoring levels). • The type of lock held on the object by this application (for object-lock levels). Mode Types: NON - No Lock IS - Intention Share Lock IX - Intention Exclusive Lock S - Share Lock SIX - Share with Intention Exclusive Lock X - Exclusive Lock IN - Intent None Z - Super Exclusive Lock U - Update Lock NS - Next Key Share Lock NX - Next Key Exclusive Lock W - Weak Exclusive Lock NW - Next Key Weak Exclusive Lock

Metric	Description
LOCK_OBJECT_TYPE	The type of object against which the application holds a lock (for object-lock-level information) or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information).
	This element can assist in determining the source of the contention for resources.
	The object type IDs are defined in *sqlmon.h*. The object may be one of the following types:
	Tablespace
	Table
	Buffer pool
	Block
	Record (or row)
	Data partition
	Internal (another type of lock held internally by the database manager)
	Automatic resize
	Automatic storage
	Values:
	AUTORESIZE_LOCK
	AUTOSTORAGE_LOCK
	BLOCK_LOCK
	EOT_LOCK
	INPLACE_REORG_LOCK INTERNAL_LOCK
	INTERNALB_LOCK
	INTERNALC_LOCK
	INTERNALJ_LOCK
	INTERNALL_LOCK
	INTERNALO_LOCK
	INTERNALQ_LOCK
	INTERNALP_LOCK
	INTERNALS_LOCK
	INTERNALT_LOCK
	INTERNALV_LOCK
	KEYVALUE_LOCK
	ROW_LOCK
	SYSBOOT_LOCK
	TABLE_LOCK
	TABLE_PART_LOCK
	TABLESPACE_LOCK
	XML_PATH_LOCK
AGENT_ID_HOLDING_LK	The agent ID of the blocker that holds the object or the application handle of the agent that holds a lock for which this application is waiting.
	The lock monitor group must be turned on to obtain this information.
	This element can assist in determining which applications are in contention for resources.
	If the value is 0 (zero) and the application is waiting for a lock, the lock is being held by an in-doubt transaction.

Metric	Description
APPL_ID_HOLDING_LK	The application ID of the application that is holding a lock on the object this application is waiting to obtain.
LOCK_WAIT_START_TIME	The date and time that this application started waiting to obtain a lock on the locked object. This element can assist in determining the severity of resource contention.
LOCK_MODE_REQUESTED	The lock mode being requested by the application. Mode Types: - NON - No Lock - IS - Intention Share Lock - IX - Intention Exclusive Lock - S - Share Lock - SIX - Share with Intention Exclusive Lock - X - Exclusive Lock - IN - Intent None - Z - Super Exclusive Lock - U - Update Lock - NS - Next Key Share Lock - NX - Next Key Exclusive Lock - W - Weak Exclusive Lock - NW - Next Key Weak Exclusive Lock
LOCK_ESCALATION	Indicates whether a lock request was made as part of a lock escalation. Use this element to help you understand the cause of deadlocks. If you experience a deadlock that involves applications performing lock escalation, you might want to increase the amount of lock memory or change the percentage of locks that any one application can request.
DATA_PARTITION_ID	The ID of the data partition for which a lock occurred. This element is only applicable to partitioned tables. A value of -1 represents a lock of the whole table and not a specific partition.
TABNAME	The name of the table. Along with TABSCHEMA, this element can assist in determining the source of the contention for resources. At the application level, application-lock level, and deadlock-monitoring level, this is the table that the application is waiting to lock, because it is currently locked by another application. For snapshot monitoring, this item is valid only when the "lock" monitor group information is turned on, and when LOCK_OBJECT_TYPE indicates that the application is waiting to obtain a table lock.
TABSCHEMA	The schema of the table. Along with TABNAME, this element can assist in determining the source of the contention for resources.

Metric	Description
TBSP_NAME	The name of a tablespace. This element can assist in determining the source of the contention for resources. It is equivalent to the TBSPACE column in the SYSCAT.TABLESPACES database catalog table. At the application level, application-lock level, and deadlock-monitoring level, this is the name of the tablespace that the application is waiting to lock. Another application currently holds a lock on this tablespace. At the lock level, this is the name of the tablespace against which the application currently holds a lock. At the tablespace level (when the buffer pool monitor group is on), this is the name of the tablespace for which information is returned. This is not returned for a table lock held on a partitioned table.
LOCK_ATTRIBUTES	The lock attributes of the lock wait. If there are no locks, the text identifier is NONE. Otherwise, it is any combination of the following, separated by a '+': • ALLOW_NEW • DELETE_IN_BLOCK • ESCALATED • INSERT • NEW_REQUEST • RR • RR_IN_BLOCK • UPDATE_DELETE • WAIT_FOR_AVAIL
LOCK_CURRENT_MODE	During a lock conversion operation, the type of lock held before the conversion is completed. For example, during an update or delete operation, it is possible to wait for an X lock on the target row. If the transaction is holding an S or V lock on the row, that would require a conversion. LOCK_CURRENT_MODE is assigned a value of S or V while the lock waits to be converted to an X lock.
LOCK_RELEASE_FLAGS	Each release flag is based on a bit flag value defined in *sqlmon.h*. The following are possible release flag settings (API constant description): SQLM_LOCKRELFLAGS_SQLCOMPILER - Locks by SQL compiler SQLM_LOCKRELFLAGS_UNTRACKED - Non-unique, untracked locks
SUBSECTION_NUMBER	Identifies the subsection associated with the returned information. This number relates to the subsection number in the access plan that can be obtained with db2expln.
LOCK_DURATION	The duration of the lock wait.

History Locks Summary Collection

The following table provides a list of the History Locks Summary metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved.
LONGEST_DURATION	A metric for internal use.
NUM_OF_EXCEEDING_LOCKS	A metric for internal use.

Instance Memory Collection

The following table provides a list of the Instance Memory metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.

POOL_ID	The pool ID:
	Values:
	  • FCMBP
	  • MONITOR
	  • OTHER

POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Instance Memory Pool Collection

The following table provides a list of the Instance Memory Pool metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.

POOL_ID	The pool ID.
	Values:
	  • FCMBP
	  • MONITOR
	  • OTHER

POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
INSTANCE_UTILIZATION	The utilization of the allocated memory from the configured memory.

Instance Memory Summary Collection

The following table provides a list of the Instance Memory Summary metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_CUR_SIZE	The total current size (in KB) of the instance memory pools per partition.
INSTANCE_UTILIZATION	The utilization of the allocated memory from the configured memory.

Instance Memory Total Collection

The following table provides a list of the Instance Memory Total metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
POOL_CUR_SIZE	The total current size (in KB) of the instance memory pools.
INSTANCE_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Memory Collection

The following table provides a list of the Database Memory metrics that are collected, and a description for each.
The memory usage of the database

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved.
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.

Metric	Description
POOL_ID	The pool ID. Values: • APP_GROUP • APPL_CONTROL • APPLICATION • BP • CAT_CACHE • DATABASE • DFM • IMPORT_POOL • LOCK_MGR • OTHER • PACKAGE_CACHE • QUERY • SHARED_SORT • SORT • STATEMENT • STATISTICS • UTILITY
POOL_SECONDARY_ID	An additional ID to help determine the memory pool for which monitoring data is returned.
POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Memory Pool Collection

The following table provides a list of the Database Memory Pool metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.

Metric	Description
POOL_ID	The pool ID. Values: • APP_GROUP • APPL_CONTROL • APPLICATION • BP • CAT_CACHE • DATABASE • DFM • IMPORT_POOL • LOCK_MGR • OTHER • PACKAGE_CACHE • QUERY • SHARED_SORT • SORT • STATEMENT • STATISTICS • UTILITY
POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Memory Pool DB Collection

The following table provides a list of the Database Memory Pool DB metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.

Metric	Description
POOL_ID	The pool ID. Values: • APP_GROUP • APPL_CONTROL • APPLICATION • BP • CAT_CACHE • DATABASE • DFM • IMPORT_POOL • LOCK_MGR • OTHER • PACKAGE_CACHE • QUERY • SHARED_SORT • SORT • STATEMENT • STATISTICS • UTILITY
POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Memory Summary Collection

The following table provides a list of the Database Memory Summary metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Memory Total Collection

The following table provides a list of the Database Memory Total metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
POOL_CUR_SIZE	The average allocated size (in KB) of a memory pool.
POOL_CONFIG_SIZE	The internally configured size (in KB) of a memory pool in the DB2 database system.
POOL_UTILIZATION	The utilization of the allocated memory from the configured memory.

Database Buffer Pools Collection

The following table provides a list of the Database Buffer Pools metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
BP_NAME	The name of the monitored buffer pool.
TABLESPACES	The number of the tablespaces using this buffer pool.
TOTAL_LOGICAL_READS	The total number of logical reads (index, XDA, and data) in the buffer pool.
TOTAL_LOGICAL_READS_S	The total number of logical reads per second (index, XDA, and data) in the buffer pool.
TOTAL_PHYSICAL_READS	The total number of physical reads (index, XDA, and data) in the buffer pool.
TOTAL_PHYSICAL_READS_S	Total number of physical reads per second (index, XDA, and data) in the buffer pool.
TOTAL_HIT_RATIO_PERCENT	The total hit ratio (index, XDA, and data reads).
DATA_LOGICAL_READS	The data logical reads, which includes access to data that is: • Already in the buffer pool when the database manager needs to process the page. • Read into the buffer pool before the database manager can process the page.
DATA_LOGICAL_READS_S	The data logical reads per second.
DATA_PHYSICAL_READS	The number of data pages read in from the tablespace containers (physical) for regular and large tablespaces.
DATA_PHYSICAL_READS_S	The number of data pages read in per second from the tablespace containers (physical) for regular and large tablespaces.
DATA_HIT_RATIO_PERCENT	The data hit ratio.

Metric	Description
INDEX_LOGICAL_READS	The index logical reads, which includes access to index pages that are: • Already in the buffer pool when the database manager needs to process the page. • Read into the buffer pool before the database manager can process the page.
INDEX_LOGICAL_READS_S	The index logical reads per second.
INDEX_PHYSICAL_READS	The number of index pages read in from the tablespace containers (physical) for regular and large tablespaces.
INDEX_PHYSICAL_READS_S	The number of index pages read in per second from the tablespace containers (physical) for regular and large tablespaces.
INDEX_HIT_RATIO_PERCENT	The index hit ratio.
XDA_LOGICAL_READS	The number of data pages for XML storage objects (XDAs) that have been requested from the buffer pool (logical) for regular and large tablespaces.
XDA_LOGICAL_READS_S	The number of data pages per second for XML storage objects (XDAs) that have been requested from the buffer pool (logical) for regular and large tablespaces.
XDA_PHYSICAL_READS	The number of data pages for XML storage objects (XDAs) read in from the tablespace containers (physical) for regular and large tablespaces.
XDA_PHYSICAL_READS_S	The number of data pages per second for XML storage objects (XDAs) read in from the tablespace containers (physical) for regular and large tablespaces.
XDA_HIT_RATIO_PERCENT	The auxiliary storage objects hit ratio.
POOL_READ_TIME_S	The total amount of time spent reading in data and index pages from the tablespace containers (physical) for all types of tablespaces.
POOL_WRITE_TIME_S	The total amount of time spent physically writing data or index pages from the buffer pool to disk.
DIRECT_READS	The number of read operations that do not use the buffer pool. Direct reads are used when: Reading LONG VARCHAR columns Reading LOB (large object) columns Performing a backup
DIRECT_READS_S	The number of read operations that do not use the buffer pool, per second.
DIRECT_WRITES	The number of write operations that do not use the buffer pool. Direct writes are used when: • Writing LONG VARCHAR columns • Writing LOB (large object) columns • Performing a restore • Performing a load • Allocating new extents for SMS tablespaces, if MPFA is enabled (which is the default)

Metric	Description
DIRECT_WRITES_S	The number of write operations that do not use the buffer pool, per second.
DIRECT_READ_TIME_S	The elapsed time (in seconds) required to perform the direct reads.
DIRECT_WRITE_TIME_S	The elapsed time (in seconds) required to perform the direct writes.
FILES_CLOSED	The total number of database files closed.
	You can use this element to assist in determining the best value for the maxfilop configuration parameter.
UNREAD_PREFETCH_PAGES	The number of pages that the prefetcher read in that were never used.
	If this number is high, prefetchers are causing unnecessary I/O by reading pages into the buffer pool that will not be used.
GBP_LOGICAL_READS	The number of times an attempt was made to read a group buffer pool (GBP) dependent page from the group buffer pool, because the page was either invalid or not present in the local buffer pool (LBP). This is for pureScale instances only, which are not yet supported.
GBP_PHYSICAL_READS	The number of times a GBP dependent page was read into the LBP from disk, because it was not found in the GBP. This is for pureScale instances only, which are not yet supported.
GBP_HIT_RATIO	The percentage of time a requested page was found in the GBP out of all of the times it was requested. If a page is found in the GBP, that saves the database time, because the database does not have to look for it in the LBP. This is for pureScale instances only, which are not yet supported.

Database Tables Collection

The following table provides a list of the Database Tables metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition from which the data was retrieved.
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
TABSCHEMA	The schema of the table.
	Along with TABNAME this element can assist in determining the source of the contention for resources.
TABNAME	The name of the table.

Metric	Description
TAB_TYPE	The type of table for which information is returned. Types: • USER_TABLE • DROPPED_TABLE - User table that has been dropped. • TEMP_TABLE - Information regarding temporary tables is returned, even though the tables are not kept in the database after being used. You may still find information about this type of table useful. • CATALOG_TABLE - System catalog table. • REORG_TABLE
TBSP_ID	An integer that uniquely represents a tablespace used by the current database.
DATA_OBJECT_PAGES	The number of disk pages consumed by a table. This represents the base table size only. Space consumed by index objects, LOB data, and long data are reported by INDEX_OBJECT_PAGES, LOB_OBJECT_PAGES, and LONG_OBJECT_PAGES, respectively.
INDEX_OBJECT_PAGES	The number of disk pages consumed by all indices defined in a table.
LOB_OBJECT_PAGES	The number of disk pages consumed by LOB data.
LONG_OBJECT_PAGES	The number of disk pages consumed by long data in a table.
XDA_OBJECT_PAGES	The number of disk pages consumed by XML storage object (XDA) data.
ROWS_READ	The number of rows read from the table.
ROWS_WRITTEN	The number of rows changed (inserted, deleted, or updated) in the table.
OVERFLOW_ACCESSES	The number of reads and writes from/to overflowed rows of this table.
PAGE_REORGS	The number of page reorganizations executed for a table.

Database Partition Backup Collection

The following table provides a list of the Database Partition Backup Collection metrics that are collected, and a description for each.

Metric	Description
DBPARTITIONNUM	The database partition number from which the data was retrieved for this row.
LAST_SUCCESSFUL_BACKUP_START	The start time of the last successful backup.
LAST_SUCCESSFUL_BACKUP_END	The end time of the last successful backup.
LAST_SUCCESSFUL_BACKUP_TYPE	The type of the last successful backup. Values: • F - Full offline • N - Full online

Metric	Description
DAYS_SINCE_LAST_SUCCESSFUL	The number of days that have passed since the last successful backup.
LAST_BACKUP_START	The start time of the last backup.
LAST_BACKUP_END	The end time of the last backup.
LAST_BACKUP_TYPE	The type of the last backup. Values: • D - Delta offline • E - Delta online • F - Offline • I - Incremental offline • N - Online • O - Incremental online
LAST_BACKUP_STATUS	The sqlcode value returned for the most recent running backup. Values: • 0 - Successful • Any other number - An error occurred For more detail, query `dbadmin db_history`.

Database Tables Global Collection

The following table provides a list of the Database Tables Global metrics that are collected, and a description for each.

Metric	Description
INSTANCE_NAME	The name of the monitored instance.
DB_NAME	The name of the monitored database.
TABSCHEMA	The schema of the table. Along with TABNAME, this element can assist in determining the source of the contention for resources.
TABNAME	The name of the table.
TAB_TYPE	The type of table for which information is returned. Types: • USER_TABLE • DROPPED_TABLE - User table that has been dropped. • TEMP_TABLE - Information regarding temporary tables is returned, even though the tables are not kept in the database after being used. You may still find information about this type of table useful. • CATALOG_TABLE - System catalog table. • REORG_TABLE
TBSP_ID	An integer that uniquely represents a tablespace used by the current database.

Metric	Description
DATA_OBJECT_PAGES	The number of disk pages consumed by a table. This represents the base table size only. Space consumed by index objects, LOB data, and long data are reported by INDEX_OBJECT_PAGES, LOB_OBJECT_PAGES, and LONG_OBJECT_PAGES, respectively.
INDEX_OBJECT_PAGES	The number of disk pages consumed by all indices defined in a table.
LOB_OBJECT_PAGES	The number of disk pages consumed by LOB data.
LONG_OBJECT_PAGES	The number of disk pages consumed by long data in a table.
XDA_OBJECT_PAGES	The number of disk pages consumed by XML storage object (XDA) data.
ROWS_READ	The number of rows read from the table.
ROWS_WRITTEN	The number of rows changed (inserted, deleted, or updated) in the table.
OVERFLOW_ACCESSES	The number of read and write operations from/to overflowed rows of this table.
PAGE_REORGS	The number of page reorganizations executed for a table.

Database Applications Collection

The following table provides a list of the Database Applications metrics that are collected, and a description for each.

Metric	Description
DB_NAME	The name of the monitored database.
DBPARTITIONNUM	The database partition from which the data was retrieved for this row.
APPL_NAME	The name of the application running on the client.
ACTIVE_TIME	The elapsed time (in seconds) spent processing an SQL request on a host database server.
FOREGROUND_BACKGROUND_TYPE	The foreground or background status of the application.
ROWS_READ	The number of rows read from the table.
ROWS_WRITTEN	The number of rows changed (inserted, deleted, or updated) in the table.
APPL_IDLE_TIME	The number of seconds since an application issued any requests to the server. This includes applications that have not terminated a transaction (for example, not issued a commit or rollback).
TOTAL_CPU	The total CPU time (in seconds) used by the application.
AGENT_USR_CPU_TIME_S	The user CPU time (in seconds) used by the application.
AGENT_SYS_CPU_TIME_S	The system CPU time (in seconds) used by the application.
DIRECT_READS	The number of direct reads by the application.
DIRECT_WRITES	The number of direct writes by the application.

Metric	Description
DIRECT_READ_TIME	The direct read time by the application.
DIRECT_WRITE_TIME	The direct write time by the application.
LOCKS_HELD	The locks held by the application.
LOCK_WAITS	The number of lock waits related to the application.
LOCK_WAIT_TIME	The lock wait time of the application.
LOCK_ESCALS	The number of lock escalations.
X_LOCK_ESCALS	The number of exclusive lock escalations.
TOTAL_SORTS	The total sorts related to the application.
TOTAL_SORT_TIME	The total sort time.
SORT_OVERFLOWS	The sort overflows of the application.
COMMIT_SQL_STMTS	The commit SQL statements related to the application.
ROLLBACK_SQL_STMTS	The rollback SQL statements related to the application.
PERCENT_SORT_OVERFLOWS	The percentage of sort operations, out of the total sort operations, that ran out of sort heap and may have required disk space for temporary storage.
FAILED_SQL_STMTS	The number of failed SQL statements.
DYNAMIC_SQL_STMTS	The number of dynamic SQL statements.
STATIC_SQL_STMTS	The number of static SQL statements.
SELECT_SQL_STMTS	The number of select SQL statements.
DDL_SQL_STMTS	The number of DDL SQL statements.
UID_SQL_STMTS	The number of DML SQL statements (UID).
ROWS_DELETED	The rows deleted by the application.
ROWS_INSERTED	The rows inserted by the application.
ROWS_UPDATED	The rows updated by the application.
ROWS_SELECTED	The rows selected by the application.
HASH_JOIN_OVERFLOWS	The hash join overflows of the application.
LOCKS_WAITING	The locks waiting related to the application.

About us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit https://www.quest.com/.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.