Monitoring Progress Database Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

The Progress database provides a multi-threaded database server that can service multiple network clients. Each server handles many simultaneous requests from clients. The server processes simple requests as a single operation to provide rapid responses; it divides complex requests into smaller tasks to minimize the impact on other users of the database.

For storage and retrieval of persistent data in an IT infrastructure, application components rely on database servers. A database server is responsible for reliably managing a large amount of data in a multi-user environment so that many users can concurrently access the same data. At the same time, a database server must also prevent unauthorized access and provide efficient solutions for failure recovery.

For ensuring high availability, performance, and security, a database server includes a wealth of data storage, caching, and retrieval functions. To ensure peak performance, a database server needs to be continuously monitored and tuned. Sometimes, there may be a sudden change in workload to the database, resulting in an increase in the number of simultaneously processed transactions. This scenario could result in a performance bottleneck at the database server. Continuous monitoring and optimization of the database server is essential for ensuring that the database server operates at its peak.

The eG Enterprise suite is programmed with a variety of tests that are designed to monitor the critical parameters of the Progress database servers. Let us now discuss how the eG Enterprise Suite performs monitoring of the Progress database server in the forthcoming chapter.

Chapter 2: How to Monitor Progress Database Server Using eG Enterprise?

eG Enterprise can monitor Progress database server in an agent-based or an agentless manner. In case of the agentless approach, the remote agent used to monitor the Progress database server should be deployed on a remote Windows host in the environment. The broad steps for monitoring the database server are as follows:

- Set the pre-requisites to configure the Progress database server to work with eG agent. See Section **2.1**.
- Manage the target Progress database server using eG Enterprise. To know how to do it, refer to Section **2.2**.
- Configuring the tests pertaining to the Progress database server. To know how to do it, refer to Section **2.3**.

2.1 Pre-Requisites for monitoring the Progress database

In order to monitor a Progress database server, a special database user account has to be created in every Progress database instance that requires monitoring. To create the database user, use the following command:

create user <user_name>, <password>

Then, for the created user, grant the following privileges one after the other.

grant select on pub."_ActAILog" to <user_name>

grant select on pub."_Area" to <user_name>

grant select on pub."_AreaStatus" to <user_name>

grant select on pub."_ActBILog" to <user_name>

grant select on pub."_Actbuffer" to <user_name>

grant select on pub."_BuffStatus" to <user_name>

grant select on pub."_Connect" to <user_name>

grant select on pub."_ActIOFile" to <user_name>

grant select on pub."_ActIndex" to <user_name>

grant select on pub."_Latch" to <user_name>

grant select on pub."_LockReq" to <user_name>

grant select on pub."_ActServer" to <user_name>

grant select on pub."_UserIO" to <user_name>

grant select on pub."_UserLock" to <user_name>

2.2 Managing the Progress Database Server

The Progress Database server cannot be automatically discovered by eG Enterprise. This implies that you will have to manually add the server into the eG Enterprise system to manage it. Follow the steps below to achieve the same:

1. Follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.
2. Next, select *Progress Database* from the **Component type** drop-down and then click the **Add New Component** button.
3. When Figure 2.1 appears, provide the **Host IP/Name** of the *Progress Database* server that you want to manage.

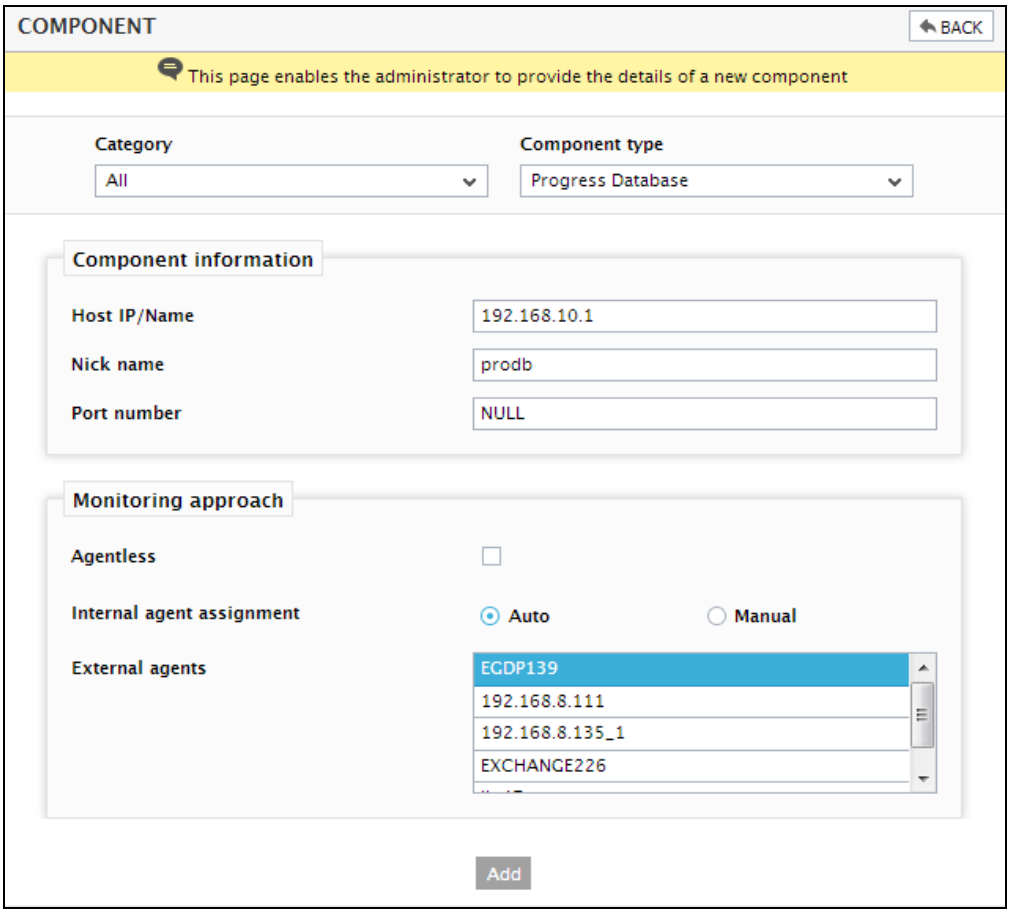

Figure 2.1: Managing a Progress Database server in an agent-based manner

4. Then, provide a **Nick name** for the server.
5. The **Port number** will be set as NULL by default. If the Progress Database server is listening on any specific port in your environment, then specify the port number here.
6. In case you are monitoring a Progress Database server in an agent-based manner, just pick an external agent from the **External agents** list box and click the **Add** button to add the component for monitoring.
7. On the other hand, if you are monitoring a Progress Database server in an agentless manner, then do the following:
   - Select the **Agentless** check box.
   - Pick the **OS** on which the Progress Database server is running.
   - Set the **Mode** to **Other**.

- Select the **Remote agent** that will be monitoring the Progress Database server. **Note that the Remote agent you choose should run on a Windows host.**
- Choose an external agent for the server by picking an option from the **External agents** list box.
- Finally, click the **Add** button to add the Progress Database server for monitoring.

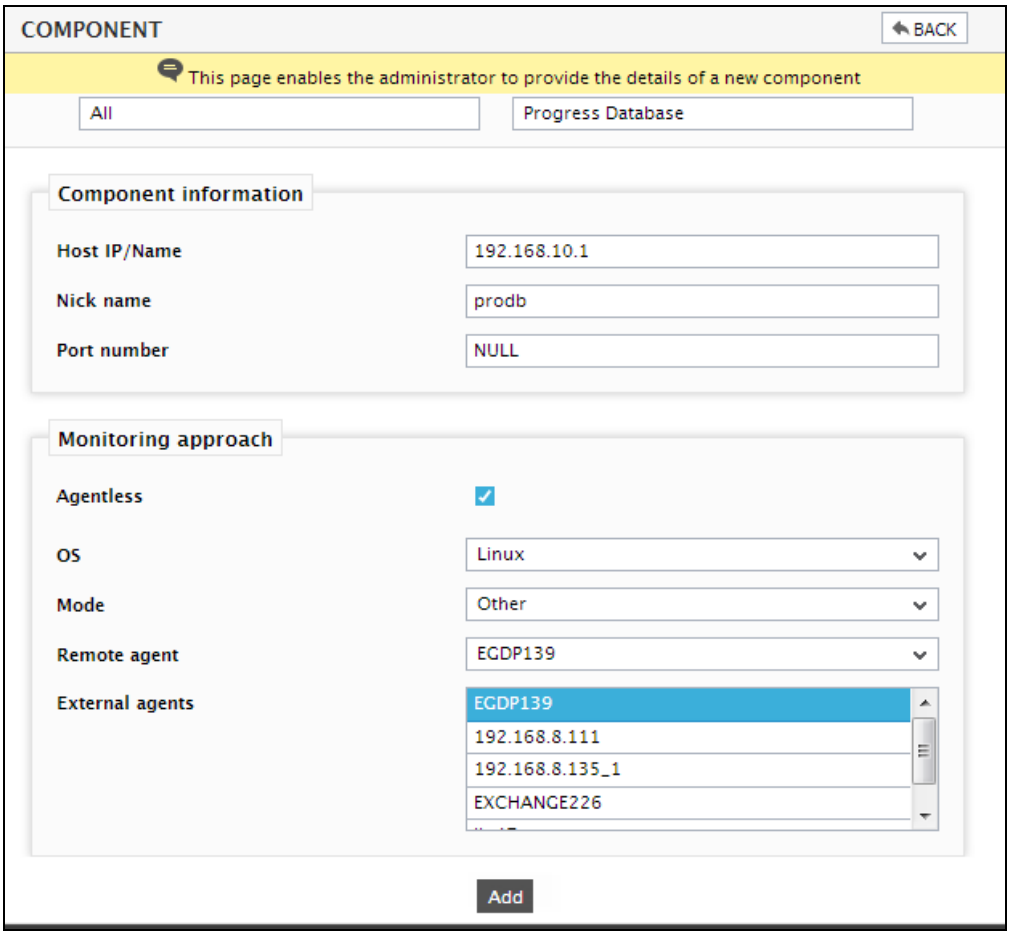

Figure 2.2: Managing a Progress Database server in an agentless manner

8. Finally, click the **Signout** button at the right, top corner of the eG admin interface to sign out.

2.3 Configuring Tests

When you try to sign out of the eG admin interface, a **LIST OF UNCONFIGURED TESTS** page will appear, revealing the list of tests mapped to the Progress Database server that require manual configuration:

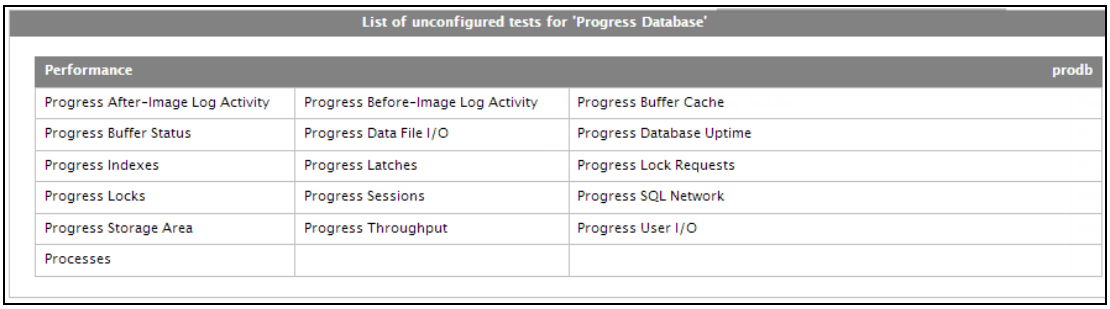

Figure 2.3: The list of unconfigured tests requiring manual configuration

Figure 2.3 indicates the list of unconfigured tests that need to be configured manually. For instance, click on the **Progress After-Image Log Activity** test. Figure 2.4 then appears listing the parameters to be configured.

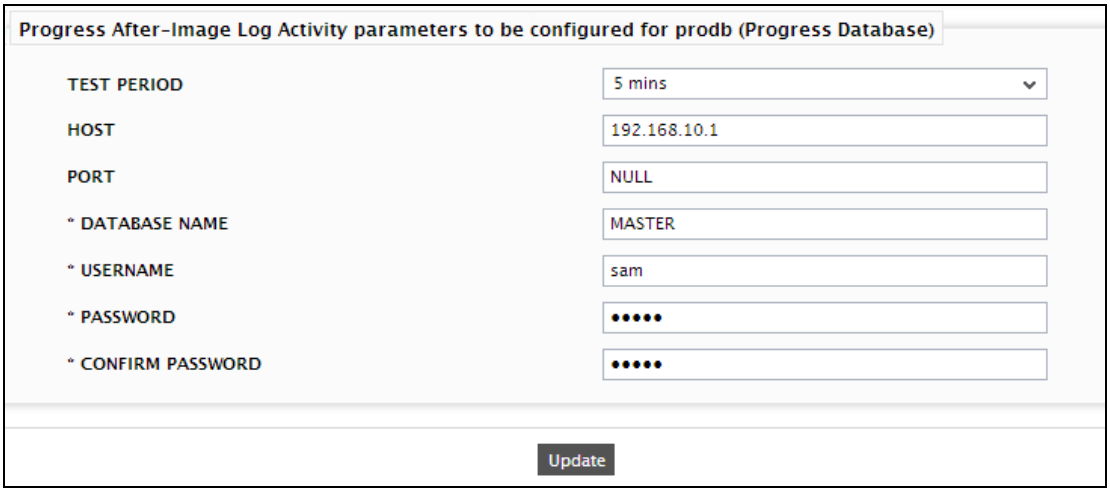

Figure 2.4: Configuring the Progress After-Image Log Activity test

To know how to configure the parameters, refer to Section **3.3.2**. Now, After configuring this test, try to signout of the eG administrative interface, this time you will be prompted to configure the **Processes** test. This test has been elaborately discussed in the *Monitoring Unix and Windows Servers* document.

After configuring the **Processes** test, sign out of the eG administrative interface. Then, login to the eG monitoring console to view the state of and metrics reported by the specialized monitoring model that eG Enterprise offers for the Progress Database server.

Chapter 3: Monitoring the Progress database servers

Any performance degradation or unavailability of the database servers can severely impact the performance of the entire service, often causing customer dissatisfaction and lost business revenue. Continuous monitoring of the Progress database servers are hence imperative.

The pre-built Progress database server monitoring model that eG Enterprise offers (see Figure 3.1), provides in-depth monitoring for the Progress database servers.

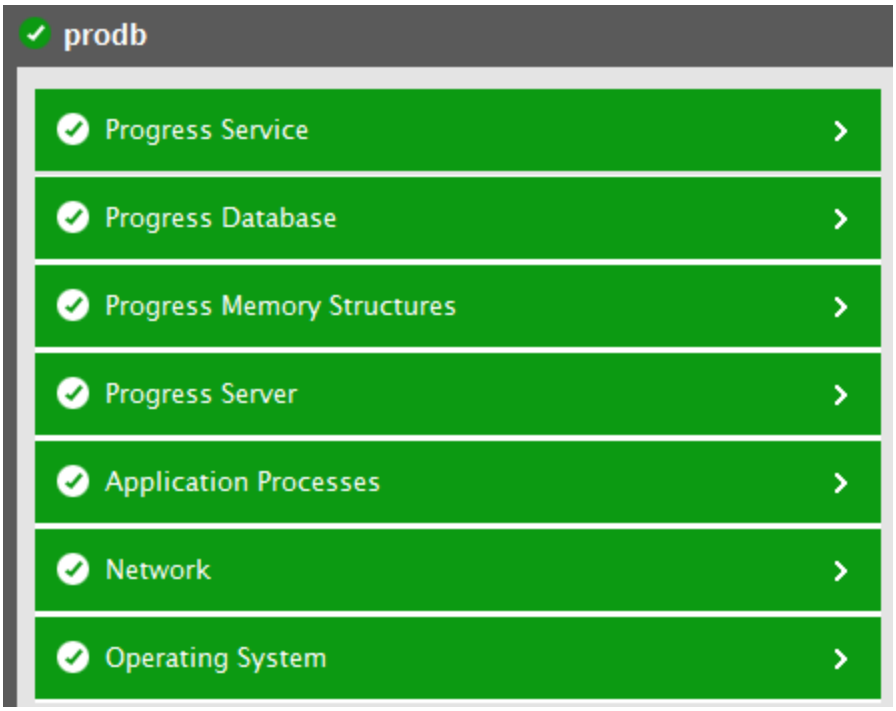

Figure 3.1: The layer model of the Progress database server

Each of the layers of this hierarchical model reports a wide variety of metrics ranging from the basic operating system-level statistics to individual database related measurements to those indicating the database engine health. Using these measurements , administrators can easily answer the following questions:

- How many writes were performed on the after-image file per second?
- How many writes were performed by the after-image writer to the after-image file per second?
- How many writes were performed on the before-image file per second?

- How many wirtes were performed by the before-image writer to the before-image file per second?
- How many time *Buffer Busy Waits* actually occurred on the before-image file?
- At what rate were the read/write operations performed on the database block of each buffer pool?
- What was the percentage of database blocks served from each buffer pool?
- What is the total number of buffers in the buffer cache of the database?
- How well the buffers in the buffer cache have been utilized?
- How many buffers are there in the page writer queue and the checkpoint queue?
- At what rate were the read/write operations performed on each datafile?
- At what rate were the data related to each datafile were read/written to the buffer?
- How long has the Progress database been up and running?
- Has the database instance been rebooted recently?
- What was the rate at which indexes were created, deleted, scanned etc?
- What is the ration of index splits to index generation?
- What is the rate at which the latches of each type were busy?
- What is the rate at which locks were held on latches of each type?
- How long was the latch of each type held?
- What is the average duration for which sessions were waiting for the latch of each type?
- What is the rate at which different locks (for e.g., EXCLUSIVE, TRANSACTION, SHARED, RECORD) were held by each user?
- How many locks were held by each user accessing the Progress database server?
- What is the rate at which the locks were being held by each user?
- How many sessions were established by each user on the Progress database server?
- At what rate were the sessions established for each user?
- What is the availability and responsiveness of the database server?
- How well each storage area is being utilized?
- What is the total amount of space allocated to each storage area?

- What is the throughput of the database server? How well data/packets were transmitted/received through the Progress database server?
- How well the read/write operations were performed on the database server by each user?
- How many reads/writes were performed by each user to the before-image file/after-image file of the database?

The *Operating System*, *Network* and *Application Processes* layers of the Progress database server monitoring model is similar to that of a Windows server model. Since these tests have been dealt with in the *Monitoring Unix and Windows Servers* document, let us now focus on the remaining layers of the Progress database server.

3.1 The Progress Server Layer

Using the tests mapped to this layer, administrators can easily track the utilization of each storage area of the target Progress database server, the uptime of the database server and the throughput of the database server.

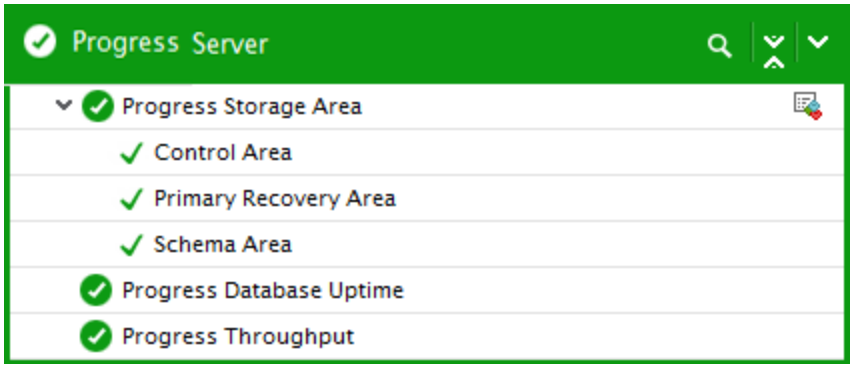

Figure 3.2: The tests associated with the Progress Server layer

3.1.1 Progress Storage Area Test

Storage areas are the largest physical unit of a database. Storage areas consist of one or more extents that are either operating system files, or some other operating system level device that is addressed randomly. A storage area is a distinct address space, and any physical address stored inside the area is generally stored relative to the beginning of the storage area.

Storage areas are identified by their names. The number and type of storage areas used varies from database to database; however, all Progress databases contains a control area, a schema area, and a primary recovery area. Certain storage areas have restrictions on the types of extents they support. The storage areas of the Progress database is briefed below:

Control Area: The control area contains only one variable-length extent: the database structure extent, which is a binary file with a .db extension.

Schema Area: The schema area can contain as many fixed-length extents as needed; however, every schema area should have a variable-length extent as its last extent.

Primary Recovery Area: The primary recovery area can contain as many fixed-length extents as needed, as long as the last extent is of variable length.

Application Data Area: The application data storage area contains all application- related database objects.

Transaction Log Area: The transaction log storage area, used for two-phase commit, uses only fixed-length extents but can use more than one. The other storage areas can use many extents but only one variable-length extent, which must be the last extent.

This test auto-discovers all the storage areas of the Progress database server and tracks the usage of each of these storage areas.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every storage area on the target Progress database server that is being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is NULL.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.

Parameter	Description
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total size	Indicates the total size of this storage area.	MB	The detailed diagnosis of this measure lists the name of the data files in the storage area, the total size of the data files, the used space of the data files, the free space of the data files and the percentage of space used by the data files. By analyzing the detailed diagnosis, the data file that is utilizing the maximum space can be determined.
Used size	Indicates the amount of storage space that is currently allocated to this storage area.	MB	

Measurement	Description	Measurement Unit	Interpretation
Free size	Indicates the amount of storage space that is currently available for use in this storage area.	MB	
Size usage	Indicates the percentage of storage space that is currently allocated to this storage area.	Percent	

3.1.2 Progress Database Uptime Test

In most production environments, it is essential to monitor the uptime of critical servers in the infrastructure. By tracking the uptime of each of the servers, administrators can determine what percentage of time a server has been up. Comparing this value with service level targets, administrators can determine the most trouble-prone areas of the infrastructure.

In some environments, administrators may schedule periodic reboots of their servers. By knowing that a specific server has been up for an unusually long time, an administrator may come to know that the scheduled reboot task is not working on a server.

The **Progress Database Uptime** test monitors the uptime of the Progress database servers.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.

Parameter	Description
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Reportmanagertime	By default, this flag is set to Yes, indicating that, by default, the detailed diagnosis of this test, if enabled, will report the shutdown and reboot times of the device in the manager's time zone. If this flag is set to No, then the shutdown and reboot times are shown in the time zone of the system where the agent is running (i.e., the system being managed for agent- based monitoring, and the system on which the remote agent is running - for agentless monitoring).
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the Onoption. To disable the capability, click on the Offoption. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Uptime	Indicates how long the database instance has been up and running.		
Uptime since last measure	Indicates the duration for which the database instance has been up since the last measurement period.	Seconds	If the value of this measure is lesser than the test frequency, it indicates that the instance was rebooted during the last measurement period.
Is rebooted?	Indicates whether the database instance was rebooted or not.		This measure reports the value *Yes* if the instance was rebooted in the last measurement period, and the value *No* if it was not rebooted. The numeric values that correspond to these measure values have been listed in the table below: **Measure Value** / **Numeric Value** Yes – 1 No – 0 **Note:** This test reports the **Measure Value**s listed in the table above to indicate whether/not the instance was rebooted. In the graph of this measure however, the same will be represented using the numeric equivalents.

3.1.3 Progress Throughput Test

This test tracks the throughput of the target Progress database server.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the target Progress database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Data transmitted	Indicates the rate at which data was transmitted from the database during the last measurement period.	MB/sec	
Data received	Indicates the rate at which data was received by the database during the last measurement period.	MB/sec	
Packets transmitted	Indicates the rate at which packets were transmitted from the database during the last measurement period.	Packets/sec	

Measurement	Description	Measurement Unit	Interpretation
Packets received	Indicates the rate at which packets were received by the database during the last measurement period.	Packets/sec	
Query requests	Indicates the rate at which query requests were serviced by the database during the last measurement period.	Requests/sec	
Transaction committed	Indicates the rate at which transactions were committed to the database during the last measurement period.	Transactions/sec	

3.1.4 Progress Waits Test

This test reports the miscellaneous wait activity on the target database server. Effective wait analysis helps determine where the database spends most of its time, and which current connections are responsible for the reported waits.

Target of the test : A Progress Database server

Agent deploying the test : An internal/external agent

Outputs of the test : One set of results for the target database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.

Parameter	Description
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Non-blocking waits for remote clients	Indicates the number of non- blocking waits started for remote clients per second.	Waits/sec	
Semaphore latch waits	Indicates the number of semaphore latch waits started per second.	Waits/sec	
Process waits for resource	Indicates the number of times the process had to wait for a resource per second.	Waits/sec	

3.1.5 Progress Table IO Test

This test reports table related activity on the target database server. Using the detailed diagnosis of this test, administrators can figure out the table that was frequently updated and the table that was read frequently.

Target of the test : A Progress Database server

Agent deploying the test: An internal/external agent

Outputs of the test : One set of results for the database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis

Parameter	Description
	measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum create rate	Indicates the maximum rate at which tables were created.	Creates/Sec	The detailed diagnosis of this measure lists the name of the table and the rate at which the table was created.
Maximum delete rate	Indicates the maximum rate at which tables were deleted.	Deletes/Sec	The detailed diagnosis of this measure lists the name of each table and the rate at which the table was deleted.
Maximum read rate	Indicates the rate at which data was read from the tables.	Reads/Sec	The detailed diagnosis of this measure lists the name of each table and the rate at which the table was read.
Maximum update rate	Indicates the rate at which data was updated in the tables.	Updates/sec	A high value indicates that there are more Update statements executed on the tables. The detailed diagnosis of this measure lists the name of each table and the rate at which the table was updated.

3.1.6 Progress Transactions Test

This test auto-discovers the users on the target Progress database server and for each user, reports the transaction activity performed on the target Progress database server. Using this test, administrators can easily track the numerical statistics of each transaction type and identify which type of transaction is executed too often on the target Progress database server and which user is responsible for those transactions.

Target of the test : A Progress Database Server

Agent deploying the test : An internal/external agent

Outputs of the test : One set of results for every user on the target database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Initiated transactions	Indicates the number of transactions initiated with the Begin status by this user.	Number	This measure is a good indicator of the current workload of the database server.

Measurement	Description	Measurement Unit	Interpretation
Active transactions	Indicates the number of transactions that are active for this user.	Number	
Outside locked transactions	Indicates the number of transactions initiated by this user that acquired locks from outside.	Number	The detailed diagnosis of this measure lists the Transaction ID, Transaction flag, Transaction star time, Duration, Connection ID, Client IP address, Connection start time, Statement type and SQL Text.
Lock unreleased transactions	Indicates the number of transactions that are completed by this user but the locks are yet to be released.	Number	
Preparing to commit transactions	Indicates the number of transactions that are being prepared for a commit by this user.	Number	
Two-phase commit transactions	Indicates the number of transactions for this user that are in two-phase commit status.	Number	Two-phase commit ensures that distributed transactions occur consistently across all databases. Two-phase commit protects against inconsistency by making sure that all databases commit the transaction, or that none commit. To ensure database integrity across all involved databases, the database engine commits database updates in two distinct phases. During the first phase, the database engine checks each database involved in a transaction to verify that it is ready to commit the transaction. During the second phase, the database engine directs the databases to commit the transaction and then verifies that they committed it properly. If there is an inconsistency, the database engine

Measurement	Description	Measurement Unit	Interpretation
			displays error messages and allows you to complete or roll back the inconsistent transaction to return the data to a consistent state.
Phase-1 transactions	Indicates the number of transactions that are entering phase -1 for this user.	Number	Phase 1 indicates that though the transaction is ready to commit it is yet to send a ready to commit response.
Phase-2 transactions	Indicates the number of transactions that are entering phase 2 status for this user.	Number	
Ready to commit transactions	Indicates the number of transactions (initiated by this user) that are ready to be committed.	Number	
Limbo transactions	Indicates the number of limbo transactions initiated by this user.		A limbo transaction (also known as an in-doubt transaction) occurs if the coordinator database commits or aborts a distributed transaction, but a hardware or software failure prevents other databases from doing likewise. This is called a limbo transaction because the processing of the transaction is temporarily suspended. A limbo transaction might occur for a variety of reasons; for example, as a result of a power outage. When a limbo transaction occurs, you must resolve the transaction to re-establish data consistency.
Active JTA transactions	Indicates the number of Java Transaction API (JTA) transactions initiated by this user that are currently active.	Number	

Measurement	Description	Measurement Unit	Interpretation
Idle JTA transactions	Indicates the number of Java Transaction API (JTA) transactions initiated by this user that are currently idle.	Number	
Prepared JTA transactions	Indicates the number of prepared JTA transactions that were initiated by this user.	Number	
Rolledback only JTA transactions	Indicates the number of JTA transactions initiated by this user were rolled back due to errors encountered.	Number	Ideally, the value of this measure should be low.
Committed JTA transactions	Indicates the number of committed JTA transactions that were initiated by this user.	Number	

3.2 The Progress Memory Structures Layer

This layer tracks the usage of the buffer cache, the I/O activity on the buffer cache, the locking activity on the database, the locks held and the latching activity on the database server.

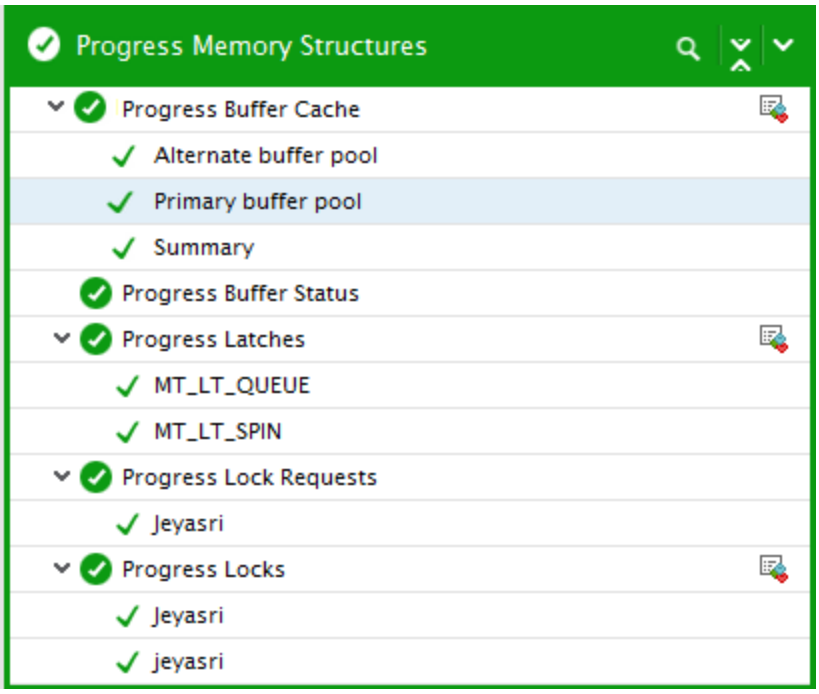

Figure 3.3: The Progress memory Structures layer

3.2.1 Progress Buffer Cache Test

The buffer pool is used to keep copies of popular database blocks in memory to minimize disk I/O. There are a number of complex data structures involved, but the most important are the buffers themselves and the buffer headers. Each buffer contains a copy of one database block and is the same size as a database block. For each buffer, there is a buffer header that describes what is in the buffer and some information about it (updated or not, which database block it is, whether the buffer is locked or not, and so forth). The buffer headers are around 150 bytes long. There are two different buffer pools in the progress database. They are:

- Primary buffer pool
- Alternate buffer pool

Whenever a user requests data from the database, for the very first time, the data is fetched from the disk and stored in the buffer pool. Subsequent requests for the same data are catered from the buffer pool. Sometimes, the buffer pool may not be able to cater the data due to unavailability of the data, buffer overhead etc leading to an increase in disk I/O and a decrease in the efficiency of the buffer pool. This may lead to over-utilization of the disk. Therefore, it is essential to monitor the buffer pools of the Progress database server. The **Progress Buffer Cache** test helps administrators in this regard!

This test auto-discovers the buffer pools in the Progress database server and for each buffer pool, this test reports how well the read/write operations were performed on the database block? In addition, this test also reports the rate at which the read/write operations were performed on the disk. Using this test administrators can figure out how well the buffer pools are utilized and optimize the buffer pools so that the disk I/O is reduced. The query response time to retrieve the data can also be improved considerably.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every buffer cache of the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Logical reads	Indicates the rate at which read operations	Reads/sec	A high value is desired for these measures. A low value is indicative

Measurement	Description	Measurement Unit	Interpretation
	were performed on the database block of this buffer pool during the last measurement period.		of ineffective cache usage, typically caused by improper cache size.
Logical writes	Indicates the rate at which write operations were performed on the database block of this buffer pool during the last measurement period.	Writes/sec	
Physical reads	Indicates the rate at which the client requests to this buffer pool directly executed on and read the database blocks from the physical disk.	Reads/Sec	Physical database reads can cause processing overheads to escalate, thereby affecting database performance. These transactions therefore have to be minimized. A high value could indicate that direct disk accesses are occurring too frequently. This in turn implies poor cache usage. You may consider resizing your cache to accommodate more number of queries, so that direct disk reads are reduced.
Physical writes	Indicates the rate at which the database blocks were directly written to the disk instead of this buffer pool.	Writes/Sec	
Hit ratio	Indicates the percentage of database blocks that were served from this	Percent	A high value is desired for this measure. A sudden/gradual decrease in the value of this

Measurement	Description	Measurement Unit	Interpretation
	buffer pool (i.e., without requiring a read from the disk).		measure is a cause of concern as this may lead to an abnormal increase in the disk I/O.

3.2.2 Progress Buffer Status Test

This test monitors the usage of the buffer memory of the Progress database server.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total buffers	Indicates the total number of buffers in the buffer cache of the database.	Number	
Used buffers	Indicates the total number of buffers that were used.	Number	
Modified buffers	Indicates the number of buffers that were modified.	Number	
Buffer usage	Indicates the percentage of buffers that were used.	Percent	A consistent value of 100 may lead to processing bottlenecks as the request processing may considerably degrade due to non availability of the buffers.
Modified buffer usage	Indicates the percentage of buffers that were modified.	Percent	
Buffers marked for checkpoint	Indicates the number of blocks that were scheduled to be written to the buffer before the end of the checkpoint.	Number	
Buffers in checkpoint queue	Indicates the number of buffers in the checkpoint queue.	Number	
Buffers in page writer queue	Indicates the number of buffers in the page writer queue.	Number	

3.2.3 Progress Latches Test

A latch is a “lightweight lock”. A latch acts like a lock, in that its purpose is to prevent data from changing unexpectedly. Just like a lock, a latch can prevent the Progress database server from accessing rows in a database, which can hurt performance. Because of this, latch wait time must be minimized.

For each type of latch on the Progress database, this test measures the latching activity.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every latch type on the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Busy rate	Indicates the number of	Latches/sec	

Measurement	Description	Measurement Unit	Interpretation
	latches of this type that were busy per second during the last measurement period.		
Lock rate	Indicates the rate at which the locks were held on this latch type during the last measurement period.	Locks/sec	A low value is desired for this measure.
Spin rate	Indicates the rate at which the process will spin again and attempt to get a latch of this type if the latch is not available immediately during the last measurement period.	Spins/Sec	
Average lock time	Indicates the time for which the latch of this type was held.	Seconds	
Average wait time	Indicates the average duration for which the sessions were waiting for latch of this type.	Seconds	Ideally, this value should be close to 0. The larger this value is, the more contention there is for latches and worse the performance of the database. If the wait time is high, check the SYSPROCESSES table of the database to see which latches are seeing most contention.

3.2.4 Progress Lock Requests Test

A Progress database server provides data concurrency and integrity between transactions using locking mechanisms. The locking activity of a database server should be monitored carefully because an application holding a specific lock for a long time could cause a number of other transactions relying on the same lock to fail.

This test indicates the level of locking activity on a database in terms of the number of locks of different modes held by each user and the time taken by each user to hold a RECORD lock and TRANSACTION lock.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every user accessing the locks on the target Progress database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Exclusive locks	Indicates the rate at which EXCLUSIVE locks were held by this user on the database during the last measurement period.	Locks/sec	Exclusive locks are used to lock data being modified by one transaction thus preventing modifications by other concurrent transactions. You can read data

Measurement	Description	Measurement Unit	Interpretation
			held by exclusive lock only by specifying a NOLOCK hint or using a read uncommitted isolation level. Because DML statements first need to read the data they want to modify you'll always find Exclusive locks accompanied by shared locks on that same data.
Record locks	Indicates the rate at which RECORD locks were held by this user on the database during the last measurement period.	Locks/sec	
Record lock waits	Indicates the rate at which this user had to wait to hold the RECORD locks on the database during the last measurement period.	Waits/sec	
Shared locks	Indicates the rate at which SHARED locks were held by this user on the database during the last measurement period.	Locks/Sec	
Transaction locks	Indicates the rate at which TRANSACTION locks were held by this user on the database during the last measurement period.	Locks/sec	
Transaction lock waits	Indicates the rate at which this user had to wait to hold the TRANSACTION locks on the database during the last measurement period.	Waits/Sec	

3.2.5 Progress Locks Test

For each user accessing on the target Progress database, this test indicates the level of locking activity in terms of the number of total locks held and total time since current lock mode was granted.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every user accessing the locks on the target Progress database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Locks count	Indicates the total number of locks that were held by this user during the last	Number	A high value may indicate one of the following: • Too many transactions

Measurement	Description	Measurement Unit	Interpretation
	measurement period.		happening • Locked resources not being released properly • Locks are being held unnecessarily.
Lock rate	Indicates the rate at which the locks were held by this user during the last measurement period.	Locks/sec	A low value is desired for this measure.

3.3 The Progress Database Layer

This layer helps administrators track the I/O activity on the Before-image files and After-image files of the target Progress database server as well as the I/O activity on each datafile. In addition, using this layer, administrators can easily identify the I/O activity of each user on the target Progress database server.

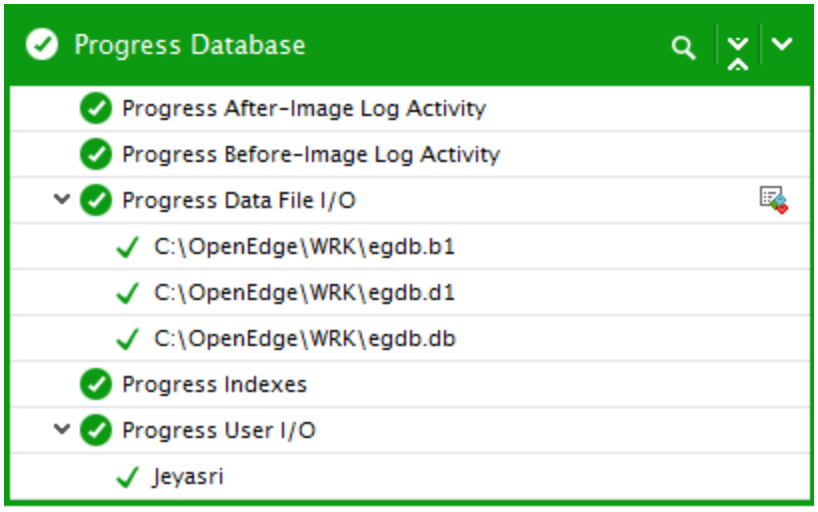

Figure 3.4: The tests associated with the Progress Database layer

3.3.1 Progress Before-Image Log Activity Test

The Before Image file(s) stores information about every transaction which makes a change to the database. If the transaction is aborted / undone, this information will be read back from the BI file to restore original values of the database back into the database if necessary.

In addition when a database goes down unexpectedly, the crash recovery process reads back and undo any transaction that was not completed.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every Progress database server that is being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total writes	Indicates the total number of writes to the before-image file during	Writes/Sec	The value of this measure includes the value of the *BIW writes* measure too.

Measurement	Description	Measurement Unit	Interpretation
	the last measurement period.		
BIW writes	Indicates the total number of writes performed by the Before- Image writer during the last measurement period.	Writes/Sec	The BIW is a background process that continually writes filled BI buffers to disk. Since writes to the BI file occur in the background, client and server processes rarely have to wait for a filled buffer to be written to disk. BIWs are optional, but highly recommended for improving I/O performance.
BIW write hit ratio	Indicates the ratio between the writes performed by the before-Image writer and the total writes to the Before image file during the last measurement period.	Percent	
Buffer busy waits	Indicates the number of times Buffer Busy waits have occurred on the before image file when a block in the buffer cache was being held by the Before image writer during the last measurement period.	Number	
Committed waits	Indicates the rate at which commit waits have occurred on the before image file when a block in the buffer cache was being held by the Before image writer during the	Records/sec	

Measurement	Description	Measurement Unit	Interpretation
	last measurement period.		
Log size	Indicates the current size of the before-image log file.	MB	
Growth rate	Indicates the rate at which the size of the before-image log file had grown during the last measurement period.	MB/sec	A consistent increase in the value of this measure is a cause for concern, as it indicates a steady growth in before- image log file size.

3.3.2 Progress After-Image Log Activity Test

The after-imaging feature lets you recover a database that was damaged when a failure caused the loss of the database or primary recovery (before image) area. When you enable after-imaging, the database engine writes notes containing a description of all database changes to the after-image (AI) files. The AI files with the roll-forward recovery process can be used to restore the database to the condition it was in before the database was lost, without losing completed transactions that occurred since the last backup. If the After-image files are not written with the changes to the database changes then and there, a considerable amount of data may be lost and hence, the database cannot be recovered completely. Also, if the buffer is not available on the after-image file, then, the processes that are initiating the writes on the file may be made to wait for a longer duration. This may also result in an increase in the wait time of the physical I/O on the file and eventual data loss if the database fails during the waiting period of the processes. Therefore, it is necessary to maintain a close watch on the writes to the AI file. The Progress After-Image Log Activity test helps administrators in monitoring the writes on the AI file.

By constantly monitoring the writes to the after-image file, this test reports the total number of writes to the file, the writes that were performed by the after-image writer and the number of times a process had to wait for the buffer in the AI file to write etc. This way, administrators can determine how well the writes were performed on the AI file and take precautionary steps to avoid data loss completely.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every Progress database server that is being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total writes	Indicates the total number of writes to the after- image file per second during the last measurement period.	Writes/Sec	The value of this measure includes the value of the *AIW writes* measure too.
AIW writes	Indicates the total number of writes performed by the after-image writer per second during the last measurement period.	Writes/Sec	
AIW write hit ratio	Indicates the ratio between the writes performed by the after-	Percent	

Measurement	Description	Measurement Unit	Interpretation
	image writer and the total writes to the after-image file during the last measurement period.		
Buffer busy waits	Indicates the number of times *Buffer Busy waits* have occurred on the after-image file when a block in the buffer cache was being held by the after- image writer during the last measurement period.	Number	The Buffer Busy wait happens when a session wants to access a database block in the buffer cache but it cannot because the buffer is busy. Another session is modifying the block and the contents of the block are in flux during the modification. The two main cases where this wait can occur are: • Another session is reading the block into the buffer • Another session holds the buffer in an incompatible mode to our request If too many buffer busy waits occur on the database, then it often indicates that the physical writes to the after-image file is taking longer to complete.
Committed waits	Indicates the rate at which commit waits have occurred on the after- image file when a block in the buffer cache was being held by the After image writer during the last measurement period.	Records/sec	
Waits for no buffer	Indicates the number of	Number	

Measurement	Description	Measurement Unit	Interpretation
	times a process had to wait since the buffer was not available in the after-image file during the last measurement period.		
Log size	Indicates the current size of the after-image log file.	MB	
Growth rate	Indicates the rate at which the size of the after-image log file had grown during the last measurement period.	MB/sec	A consistent increase in the value of this measure is a cause for concern, as it indicates a steady growth in after-image log file size.

3.3.3 Progress Data File I/O Test

If a datafile is able to process I/O requests to it quickly, it is a sign of the good health of the Progress database server. On the other hand, any slowdown in IOPS could indicate a serious processing bottleneck on the server, probably caused by a poor indexing engine or badly structured tables in a datafile. Administrators should hence continuously track the reads/writes to every datafile on the Progress database server, identify the rate at which data related to the datafile was read/written from the buffer, and measure how well the requests were processed by the datafile. For this purpose, you can run the **Progress Data File IO** test.

This test auto-discovers the datafiles on the Progress database server and reports the rate at which reads/writes were performed on the datafile. This test also reports the rate at which data was read/written from the buffer and the rate at which data is read/written from the disk directly. In the process, I/O processing bottlenecks can be detected and the datafiles that are affected can be identified.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every data file of the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Reads	Indicates the rate at which read operations were performed on this datafile during the last measurement period.	Reads/Sec	
Writes	Indicates the rate at which write operations were performed on this datafile during the last measurement period.	Writes/Sec	
Buffer reads	Indicates the rate at which data related to this datafile were read from the buffer during the last	Reads/Sec	

Measurement	Description	Measurement Unit	Interpretation
	measurement period.		
Buffer writes	Indicates the rate at which data related to this datafile were written on the buffer during the last measurement period.	Writes/Sec	
Unbuffer reads	Indicates the rate at which data related to this datafile was read from the disk instead of the buffer during the last measurement period.	Reads/Sec	
Unbuffer writes:	Indicates the rate at which data related to this datafile was directly written to the disk during the last measurement period.	Writes/Sec	
Unbuffer read ratio	Indicates the ratio of read operations served from the disk to the total read operations for this datafile.	Percent	A low value is desired for this measure. A sudden/gradual increase in the value of this measure implies that the read and write operations are performed on the disk.

3.3.4 Progress Indexes Test

An index is a data structure that a database uses to reduce the amount of time it takes to perform certain operations. An index can also be used to ensure that duplicate values don't appear where they are not needed.

This test monitors the indexes on the Progress database server and helps administrators quickly and accurately assess the effectiveness of these indexes.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the target Progress database server that is to be monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Index create rate	Indicates the rate at which indexes were created during the last measurement period.	Creates/Sec	
Index delete rate	Indicates the rate at which index entries were deleted during the last measurement period.	Deletes/Sec	
Index scan rate	Indicates the rate at which index entries were	Scans/Sec	An index scan occurs when the database manager accesses an

Measurement	Description	Measurement Unit	Interpretation
	scanned (searched for) during the last measurement period.		index for any of the following reasons: • To narrow the set of qualifying rows (by scanning the rows in a certain range of the index) before accessing the base table. • To order the output. • To retrieve the requested column data directly. If all of the requested data is in the index, the indexed table does not need to be accessed. This is known as an index-only access. Typically, a high value of this measure is desired, as index scans are I/O-friendly operations. A high value is desired for this measure. If the value is low or falls consistently, it indicates bottlenecks while performing index scans.
Index split rate	Indicates the rate at which index entries were split during the last measurement period.	Splits/Sec	
Index split ratio	Indicates the ratio of index splits to the index generation during the last measurement period.	Percent	

3.3.5 Progress User I/O Test

This test auto-discovers the users on the target Progress database server and periodically reports the read-write activity of each user. Using this test, administrators can easily identify which user is performing resource intensive I/O operations. This way, this test reveals the irregularities in the I/O operations performed by the users and further analyze the real reason behind the abnormally high I/O operations performed by the user.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every user on the target Progress database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Access rate	Indicates the rate at which database access operations were performed	Accesses/Sec	

Measurement	Description	Measurement Unit	Interpretation
	by this user during the last measurement period.		
Reads	Indicates the rate at which read operations were performed on the database by this user during the last measurement period.	Reads/Sec	
Writes	Indicates the rate at which write operations were performed on the database by this user during the last measurement period.	Writes/Sec	
Before-image reads	Indicates the rate at which read operations were performed on the Before-image files by this user during the last measurement period.	Reads/Sec	
Before-image writes	Indicates the rate at which write operations were performed on the Before-image file by this user during the last measurement period.	Writes/Sec	
After-image reads	Indicates the rate at which read operations were performed on the After-image files by this user during the last measurement period.	Reads/Sec	
After-image writes	Indicates the rate at which write operations were performed on the After-image files by this user during the last measurement period.	Writes/Sec	

3.3.6 Progress Unused Indexes Test

While at one end indexes greatly enhance database performance, at the other they also add significant overhead to table change operations. Useless/unused indices can therefore be unnecessary resource hogs. Such indexes are typically not used by any regular query and may not enforce a constraint. However, these unneeded indexes cost you in several ways: they slow updates, inserts and deletes; they may keep HOT from updating the row in-place, they add to query planning time; they take time to backup and restore. Administrators hence need to identify such indexes and eliminate them. The **Progress Unused Indexes** test helps administrators achieve the same!

This test reports the number and names of unused/useless indexes, and thus prompts administrators to remove them so as to save the server from unnecessary performance degradations.

Target of the test : A Progress Database server

Agent deploying the test: An internal/external agent

Outputs of the test : One set of results for the target database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an

Parameter	Description
	optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Unused indexes	Indicates the number of indexes that are currently unused/useless on the server.	Number	A high value of this measure is a cause for concern. Use the detailed diagnosis of this measure to know which indexes in a database are unused. The detailed diagnosis also reveals how often each such index has been updated, so that you can assess the unnecessary overheads that the database may have incurred in the process.

3.4 The Progress Service Layer

This layer tracks the overall health of the service offered by the database server to clients. This layer helps administrators track the availability and responsiveness of the database server. In addition, this layer helps administrators to track session-level information regarding the usage of the database server.

Figure 3.5: The tests associated with the Progress Service layer

3.4.1 Progress Sessions Test

In the database context, the connection between the user process and the server process is called a session. The server process communicates with the connected user process and performs tasks on behalf of the users.

This test indicates the level of activity on a database in terms of the number of sessions initiated by each user and the rate at which the sessions are initiated.

Target of the test : A Progress Database server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each user on the target Progress Database server that is being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.

Parameter	Description
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On**option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total sessions	Indicates the total number of sessions established by this user on the database.	Number	A high value may indicate that the user is causing a high load on the server. The detailed diagnosis of this measure if enabled, lists all the current sessions of the user along with the Connection PID, Client IP Address, Connection Client Type, Connection Type and Connection Time.
Session rate	Indicates the rate at which sessions were established by this user on the database during the last measurement period.	Sessions/sec	Comparing the value of this measure will point you to the user who has established the maximum number of sessions.

3.4.2 Progress Users Test

A Progress database server provides data concurrency and integrity between transactions using locking mechanisms. The locking activity of a database server should be monitored carefully because an application holding a specific lock for a long time could cause a number of other transactions relying on the same lock to fail.

This test indicates the level of locking activity on a database in terms of the number of locks of different modes held by each user per second and the time taken by each user to hold a RECORD lock and TRANSACTION lock. In addition, this test reveals the number of total locks held and total time since current lock mode was granted. This test also periodically reports the read-write activity of each user. Using this test, administrators can easily identify which user is performing resource intensive I/O operations. This way, this test reveals the irregularities in the I/O operations performed by the users and further analyze the real reason behind the abnormally high I/O operations performed by the user.

Target of the test : A Progress Database server

Agent deploying the test : An internal/external agent

Outputs of the test : One set of results for every user on the target Progress database server being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.

Parameter	Description
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Exclusive locks	Indicates the rate at which EXCLUSIVE locks were held by this user on the database during the last measurement period.	Locks/Sec	Exclusive locks are used to lock data being modified by one transaction thus preventing modifications by other concurrent transactions. You can read data held by exclusive lock only by specifying a NOLOCK hint or using a read uncommitted isolation level. Because DML statements first need to read the data they want to modify you'll always find Exclusive locks accompanied by shared locks on that same data.
Record locks	Indicates the rate at which RECORD locks were held by this user on the database during the last measurement period.	Locks/Sec	
Record lock waits	Indicates the rate at which this user had to wait to get hold of RECORD locks on the database during the last measurement period.	Waits/sec	
Shared locks	Indicates the rate at which SHARED locks were held by this user on the database during the last measurement period.	Locks/sec	
Transaction locks	Indicates the rate at which TRANSACTION locks	Locks/sec	

Measurement	Description	Measurement Unit	Interpretation
	were held by this user on the database during the last measurement period.		
Transaction lock waits	Indicates the rate at which this user had to wait to get hold of TRANSACTION locks on the database during the last measurement period	Waits/sec	
Locks count	Indicates the total number of locks that were currently held by this user.	Number	A high value may indicate one of the following: • Too many transactions happening • Locked resources not being released properly • Locks are being held unnecessarily The detailed diagnosis of this measure lists the Table name, Lock type, Userlock flag, Record ID, Transaction ID, Transaction flag, Transaction state, Transaction start time, Duration, Client IP address, Statement type and SQL text.
Lock rate	Indicates the rate at which the locks were held by this user during the last measurement period.	Locks/sec	A low value is desired for this measure.
Current sessions	Indicates the total number of sessions established by this user on the database.	Number	A high value may indicate that the user is causing a high load on the server. The detailed diagnosis of this measure if enabled, lists all the current sessions of the user along with the Connection

Measurement	Description	Measurement Unit	Interpretation
			PID, Client IP Address, Connection Client Type, Connection Type ,Connection Time, Statement type and SQL text.
Session usage	Indicates the session utilization of this user.	Percent	A high value may indicate that the user is causing a high load on the server. By increasing the _Startup-MaxUsers Parameter the maximum number of sessions allowed for the user can be increased. It will avoid the failure of session's requests on the server.
Access rate	Indicates the rate at which this user performed database access operations during the last measurement period.	Accesses/sec	
Reads	Indicates the rate at which read operations were performed by this user on the database server during the last measurement period.	Reads/sec	
Writes	Indicates the rate at which data was written to the database server by this user during the last measurement period.	Writes/sec	
Before-image reads	Indicates the rate at which data was read from the Before-Image file by this user during the last measurement period.	Reads/sec	
Before-image writes	Indicates the rate at which data was written to the Before-Image file by this user during the last measurement period.	Writes/sec	

Measurement	Description	Measurement Unit	Interpretation
After-image reads	Indicates the rate at which data was read from the After-Image file by this user during the last measurement period.	Reads/sec	
After-image writes	Indicates the rate at which data was written to the After-Image file by this user during the last measurement period.	Writes/sec	

3.4.3 Progress SQL Network Test

Using the JDBC API, this test reports the availability and responsiveness of the server, and collects statistics pertaining to the traffic into and out of the database server.

Target of the test : A Progress Database server

Agent deploying the test : An external agent; if you are running this test using the external agent on the eG manager box, then make sure that this external agent is able to communicate with the port on which the target Oracle server is listening. Alternatively, you can deploy the external agent that will be running this test on a host that can access the port on which the target Oracle server is listening.

Outputs of the test : One set of results for Progress database server instance that is being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.

Parameter	Description
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Database availability	Whether the database instance is responding to requests.	Percent	The availability is 100% when the instance is responding to a request and 0% when it is not. Availability problems may be caused by a misconfiguration/malfunctioning of the database instance, or because the instance is using an invalid user account. Besides the above, this measure will report that the server is unavailable even if a connection to the database instance is unavailable, or if a query to the database fails. In this case, you can check the values of the *DB connection availability* and *Query processor availability* measures to know what is exactly causing the database instance to not respond to requests - is it owing to a connection unavailability? or is it due to a query failure? Although included as part of the **Oracle SQL Network** test, this measure maps to the **Oracle Service**

Measurement	Description	Measurement Unit	Interpretation
			layer.
Database response time	The time taken by the database to respond to a user query. This is the sum total of the connection time and query execution time.	Secs	A sudden increase in response time is indicative of a bottleneck at the database server. This could even be owing to a connection delay and/or long running queries to the database. Whenever the value of this measure is high, it would be good practice to compare the values of the Connection time and Query execution time measures to zero-in on the root-cause of the poor responsiveness of the server - is it because of connectivity issues? or is it because of inefficient queries? Although included as part of the **Oracle SQL Network** test, this measure maps to the **Oracle Service** layer.
Connection availability	Indicates whether the database connection is available or not.	Percent	If this measure reports the value 100 , it indicates that the database connection is available. The value 0 on the other hand indicates that the database connection is unavailable. A connection to the database may be unavailable if the database is down or if the database is listening on a port other than the one configured for it in the eG manager or owing to a poor network link. If the *Database availability* measure reports the value 0, then, you can check the value of this measure to determine whether/not it is due to the unavailability of a connection to the server.

Measurement	Description	Measurement Unit	Interpretation
Connection response time	Indicates the time taken by the database connection.	Secs	A high value could indicate a connection bottleneck. Whenever the Total response time of the measure soars, you may want to check the value of this measure to determine whether a connection latency is causing the poor responsiveness of the server.
Query availability	Indicates whether the database query is executed successfully or not.	Percent	If this measure reports the value 100, it indicates that the query executed successfully. The value 0 on the other hand indicates that the query failed. In the event that the *Database availability* measure reports the value 0, check the value of this measure to figure out whether the failed query is the reason why that measure reported a server unavailability.
Query response time	Indicates the time taken for query execution.	Secs	A high value could indicate that one/more queries to the database are taking too long to execute. Inefficient/badly designed queries to the database often take too long to execute. If the value of this measure is higher than that of the *Connection time* measure, you can be rest assured that long running queries are causing the respond slowly to requests.
Number of records	Indicates the number of records fetched from the database.	Number	The value 0 indicates that no records are fetched from the database.

3.4.4 Progress Long Running Queries Test

For each user auto-discovered on the target database server, this test tracks the currently executing queries and determines the number of queries that have been running for a long time. This test also tracks the maximum time taken to execute the queries. You can also use the detailed diagnosis of this test to drill down to the exact queries that have been running for an unreasonably long time, and thus isolate the user executing the resource-intensive queries to the database.

Target of the test : A Progress Database server

Agent deploying the test : An internal/external agent

Outputs of the test : One set of results for each user on the target database server

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed
Host	The IP address of the Progress database server.
Port	The port number on which the database server is listening. By default, this is *NULL*.
Database Name	Specify the name of the Progress database instance that is to be monitored.
Username	In order to monitor a Progress database instance, a special database user account has to be created in every Progress database instance that requires monitoring. This special user needs to be granted a set of privileges. To know how to create the database user and grant the required privileges, refer to Section **2.1**. Specify the name of such a user in this text box.
Password	The password of the specified Username.
Confirm Password	Confirm the password by retyping it here.
Elapsed Time	In the Elapsed Time text box, specify the duration (in seconds) beyond which a query should have executed for it to be regarded as a long running query. The default value is 10.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are

Parameter	Description
	detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurement made by the test

Measurement	Description	Measurement Unit	Interpretation
Long running queries	Indicates the number of queries currently executing (for this user) on the database server that have been running for more time than the configured Elapsed Time.	Number	The detailed diagnosis for this measure indicates the exact queries and which user is executing the queries. This information can be very useful in identifying queries that may be candidates for optimization. By comparing the value of this measure across the users, administrators can identify the user who is executing the most number of resource intensive queries.
Max elapsed time	Indicates the maximum duration taken to execute the queries by this user.	Seconds	

4

Chapter 4: Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to the **Progress** database servers. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.