Benchmark Factory® 8.0.1

User Guide

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

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

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

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

Patents

This product includes patent pending technology. For the most current information about applicable patents for this product, please visit our website at www.quest.com/legal.

Trademarks

Quest, Quest Software, Benchmark Factory, Foglight, Spotlight, SQL Navigator, Toad, SharePlex, and the Quest logo are trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at www.quest.com/legal. Microsoft, Windows, Windows Server, Windows Vista, Visual Studio, SQL Server, SharePoint, Access and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Oracle is a trademark or registered trademark of Oracle and/or its affiliates in the United States and other countries. Citrix® and XenApp™ are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. SAP is the registered trademark of SAP AG in Germany and in several other countries. IBM and DB2 are registered trademarks of International Business Machines Corporation. All other trademarks, servicemarks, registered trademarks, and registered servicemarks are property of their respective owners.

Benchmark Factory
User Guide
Wednesday, November 08, 2017

Contents

1

New in this Release

Benchmark Factory 8.0.1

Benchmark Factory 8.0.1 is a maintenance release and includes resolved issues and minor enhancements. See the *Benchmark Factory 8.0.1 Release Notes* for a complete list.

New Features in 8.0

Benchmark Factory® 8.0 includes the following new features and enhancements.

Enhanced REST API

With this release the Benchmark Factory REST API has been enhanced to allow users more control of jobs, tests, test runs, etc. This includes the ability to access the transactions in tests, agents used by jobs, test results of a test run, as well as the Benchmark Factory settings.

Support for JSON format. Responses can now be returned in JSON format. To return data using the JSON format, specify JSON in the request Accept header. Both XML and JSON are supported response or request formats.

Access to Additional REST Resources. Additional resources have been added to the Benchmark Factory REST API. For a complete list, see the Benchmark Factory Help. The Help also includes a useful description of the REST resource structure and the URI hierarchy.

Integrate Benchmark Factory into Continuous Testing Process

This release includes additional features that allow you to easily integrate Benchmark Factory into your Continuous Integration or Continuous Testing process.

BMFServer.exe

This release includes a non-UI Benchmark Factory application, BMFServer.exe.

- BMFServer.exe is installed into the **bin** sub-directory of the installation directory when you install Benchmark Factory.
- To start BMFServer.exe, go to the installation directory. The default location is C:\Program Files\Quest Software\Benchmark Factory <version>\bin. Then double-click **BMFServer.exe**.
- BMFServer.exe performs the same functionality as BFactory.exe, except it has no graphic user interface.

- Use BMFServer.exe when automating your continuous improvement/testing process. You can run BMFServer.exe using a script, a custom application, or the Command Prompt window. BMFServer.exe can be used with the Benchmark Factory REST API.
- You cannot run the Benchmark Factory console and BMFServer.exe at the same time on the same machine.

BMFAgent.exe

Benchmark Factory now includes a non-UI Agent, BMFAgent.exe.

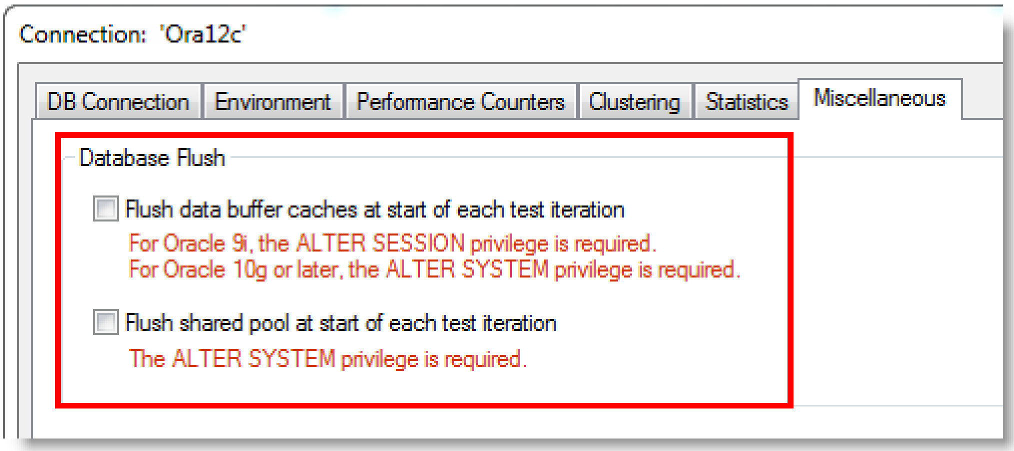
- BMFAgent.exe is installed into the **bin** sub-directory of the installation directory at the time you install Benchmark Factory.
- BMFAgent.exe has the same functionality as Agent.exe, except BMFAgent.exe has no graphic user interface.
- To modify BMFAgent.exe settings, use the BMFAgent.ini file located in the ProgramData\Quest Software\BMF directory or use the Command Prompt window. Settings you can modify include the console machine name/IP and the console port number.
- You can run multiple instances of BMFAgent.exe at the same time on the same machine.

Standard Benchmark Testing

Clear Cached Data between Iterations

You can now instruct Benchmark Factory to clear cached data between test iterations. Because cached data can improve performance, this new feature can prevent cached data from affecting subsequent iterations.

- **Oracle**—For Oracle connections, you can instruct Benchmark Factory to clear data buffer caches or shared pool (or both). To specify these options, select **Edit | Connections**. Then select a connection and click the **Edit** button. In the connection properties dialog, select the **Miscellaneous** tab. Then specify the **Database Flush** options for this connection.


Connection: 'Ora12c'

Database Flush

□ Flush data buffer caches at start of each test iteration

For Oracle 9i, the ALTER SESSION privilege is required.
For Oracle 10g or later, the ALTER SYSTEM privilege is required.

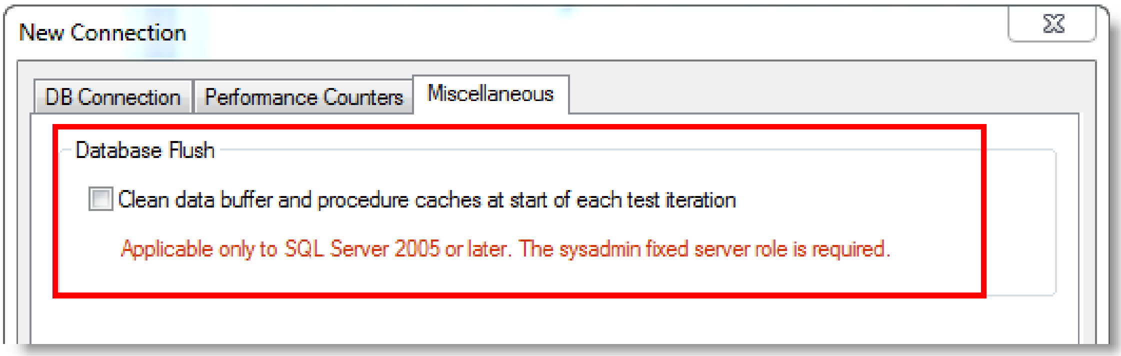
□ Flush shared pool at start of each test iteration

The ALTER SYSTEM privilege is required.

Tip: You can also specify default values to automatically apply to Oracle connections in **Edit | Settings | Oracle**.

Note: Specific database privileges are required to perform this action. The privilege required is dependent on the database version and the option selected. See the online Help for more information.

- **SQL Server®**—For SQL Server connections, you can instruct Benchmark Factory to clear data buffer caches and procedure caches. To specify this option, select **Edit | Connections**. Then select a connection and click the **Edit** button. In the connection properties dialog, select the **Miscellaneous** tab. Then specify the **Database Flush** option for this connection.



Database Flush

□ Clean data buffer and procedure caches at start of each test iteration

Applicable only to SQL Server 2005 or later. The sysadmin fixed server role is required.

Tip: You can also specify a default value to apply to SQL Server connections in **Edit | Settings | SQL Server**.

Note: This option is only applicable to SQL Server 2005 or later. The sysadmin fixed server role is required.

Connections

Microsoft® SQL Server 2017. This release includes support for SQL Server 2017. Benchmark Factory has been tested against SQL Server 2017 running on Windows or Linux.

IBM® DB2®. This release includes support for IBM DB2 11.1 for LUW and for z/OS.

General

Adding Bind Parameters

It is now easier to add a bind parameter/value pair in **Test Options | Transactions**.

- When adding a new statement in the **Add SQL Transaction** dialog or when editing an existing statement, you can now simply double-click within the **Bind Parameters** tab to add a bind parameter and parameter value.

Installation

Universal C Runtime Component

The Universal C Runtime component for Windows is required. See Universal C Runtime Update or Visual C++ Redistributable for Visual Studio 2015 to download this software.

Note: If you encounter an error when installing this software, install the missing prerequisite software. For Windows 8.1 or Windows Server 2012 R2, install the April 2014 update: https://support.microsoft.com/en-us/kb/2919355. For other operating systems, see the Universal C Runtime Update Prerequisites section.

Note: The features described in the New in This Release section apply to the commercial version of Benchmark Factory and may not be available in the freeware edition.

2

Welcome to Benchmark Factory

Benchmark Factory® is a database performance testing tool that allows you to conduct database workload replay, industry-standard benchmark testing, and scalability testing.

Review the following topics for a quick overview of Benchmark Factory.

- About Benchmark Factory
- Understanding Benchmark Factory Testing
- Understanding Benchmark Factory Terminology
- How Benchmark Factory Works
- The Benchmark Factory Environment
- Getting Started - the Benchmark Factory Workflow

Additional Resources

Learn More and Connect with the Community

Use the following links to videos, blogs, and community resources to learn more about Benchmark Factory, to find answers to questions, and to advance your knowledge through Benchmark Factory tips and tricks.

- Benchmark Factory Community—Ask questions, join the discussion, find videos and user guides, and join the Beta program (when available).
- Release Notes and Product Documentation (Benchmark Factory Community)
- Videos and Tutorials—Find video demos of Benchmark Factory features.
- Benchmark Factory Blog—Find How To's, useful articles, and tips and tricks for using Benchmark Factory.
  - How to use the REST API (part 1)— Learn how to use the Benchmark Factory REST API.
  - Benchmarking Best Practices—10 best practices to help you get started with database benchmarking.
- Product Documentation —Find documents for current and past releases, as well as documents for other Toad products, on the Quest Support Portal.
- Toad World—Visit other Toad Communities, including Toad for Oracle. Find DBMS and SQL knowledge, find software downloads, and find answers to your database questions.
- Benchmark Factory Product Information—Find white papers, product demos, and purchasing information.

3

Overview of Benchmark Factory

About Benchmark Factory

Benchmark Factory is a database performance and code scalability testing tool that simulates users and transactions on the database and replays production workload in non-production environments. With Benchmark Factory developers, DBAs, and QA teams can validate that their databases will scale as user loads increase, application changes are made, and platform changes are implemented.

Benchmark Factory is available for Oracle®, SQL Server®, IBM® DB2®, SAP®, MySQL, and other databases via ODBC connectivity.

What You Can Do With Benchmark Factory

Benchmark Factory places enormous stress on a database system, which is typically hard to achieve in a standard testing environment. A system typically breaks under extreme load. By identifying system capacity and performance bottlenecks before they occur, Benchmark Factory facilitates proactive testing, which in turn reduces downtime, development costs, and potential loss of revenue.

Benchmark Factory allows you to:

- Determine system throughput and capacity for database systems
- Simulate thousands of concurrent users with a minimal amount of hardware
- Find applications that do not scale well with an increase in the number of users
- Find breaking points, weak links, or bottlenecks of a system
- Quantify application or server performance

The Benchmark Factory Environment

Benchmark Factory consists of the following components:

Benchmark Factory Console (Desktop Application)

The Benchmark Factory console is used to create, run, and review Jobs (tests). You can also use the console to deploy agents, access the Repository, access reports, and create connections.

The Repository

All test results are collected and stored in the repository for data analysis and reporting. Benchmark Factory collects a vast amount of statistics, including overall server throughput (measured in transactions per second, bytes transferred, etc.) and detailed transaction statistics by individual workstation producing a load. You use these statistics to measure, analyze, and predict the capacity of a system.

The test results stored in the repository are displayed when you view the Results tab for a selected job or when you use the Run Reports component (Tools | Run Reports).

Agents

Agents simulate virtual users and send transactions to the system-under-test (database).

Run Reports

Benchmark Factory Run Reports is a separate component used to view the detailed test results in a report format. You can open Run Reports from the Benchmark Factory console (Tools | Run Reports) or from the Start menu (Benchmark Factory | Run Reports).

Integration with Other Toad Products

Benchmark Factory integrates with other Toad products to allow IT departments to quickly measure the capacity and performance of their systems to ensure that users will experience fast response times. For instance, Benchmark Factory can run multiple workloads in conjunction with Spotlight™ products. This provides the ability to detect and diagnose issues, allowing users to resolve bottlenecks, slow performance, and application flaws before an application is entered into production.

Understanding Benchmark Factory Testing

Review the following for a better understanding of how Benchmark Factory performs benchmark testing on your database (system-under-test).

Agents, Virtual Users, and Automated Testing

When you load test with Benchmark Factory, you are performing automated testing. Automated testing is the process of using software to test computer hardware or software implying that the software is used instead of actual users. The software acts as a "virtual" user. For example, say you would like to test your database with two hundred virtual users over a given period of time. Benchmark Factory lets you select two hundred virtual users and the length of time you wish to perform the test.

Benchmark Factory uses Agents to deploy the virtual users. An Agent is a software routine that waits in the background and performs an action when a specified event occurs. One Agent can simulate thousands of virtual users (limited by hardware and workload characteristics) at a time. Each virtual user has their own connection to the system under test.

Understanding Benchmarks

A benchmark is a performance test of hardware and/or software on a system-under-test. Benchmark Factory provides the option of using industry standard benchmarks during the load testing process. Benchmarks measure system peak performance when performing typical operations.

Benchmark Factory comes equipped with the following industry standard benchmarks:

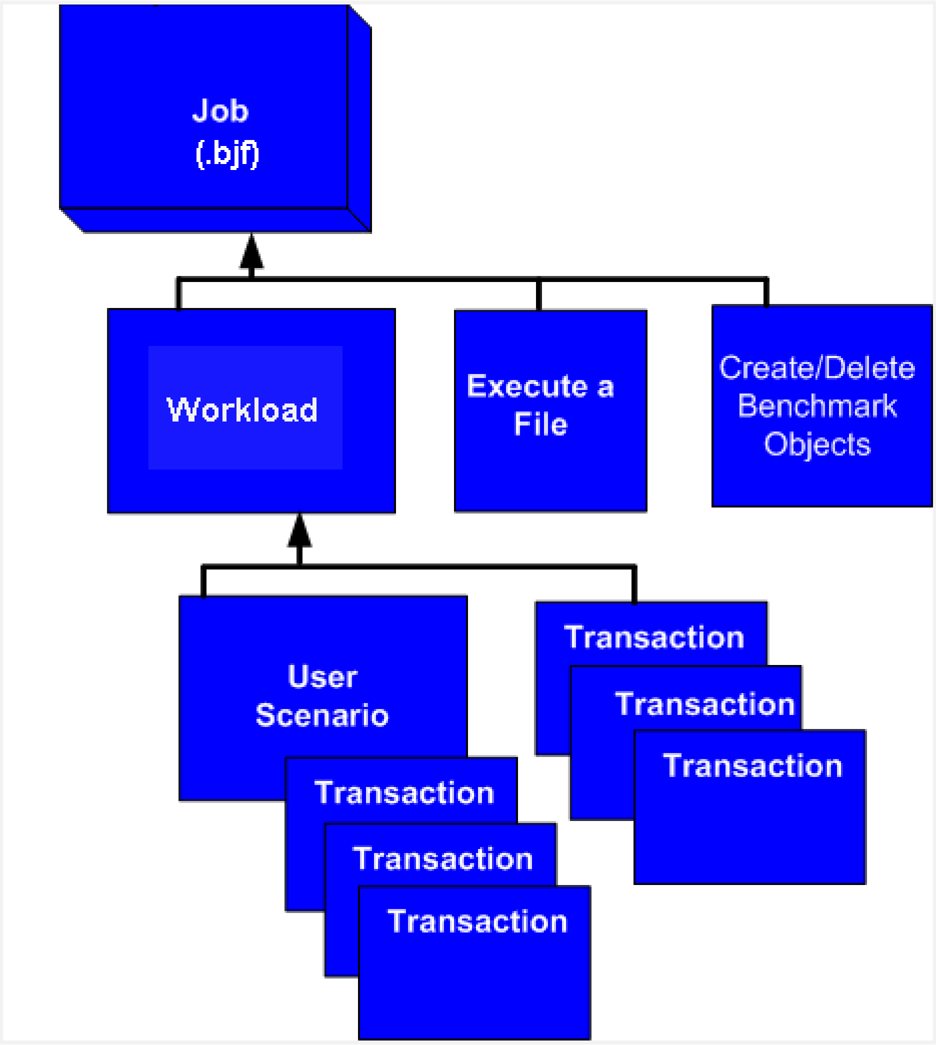
- AS3AP Benchmark
- Scalable Hardware Benchmark
- TPC-B Benchmark
- TPC-C Benchmark
- TPC-D Benchmark

- TPC-E Benchmark
- TPC-H Benchmark

Understanding Benchmark Factory Terminology

The following provides a list of terminology required to understand the Benchmark Factory load testing process.

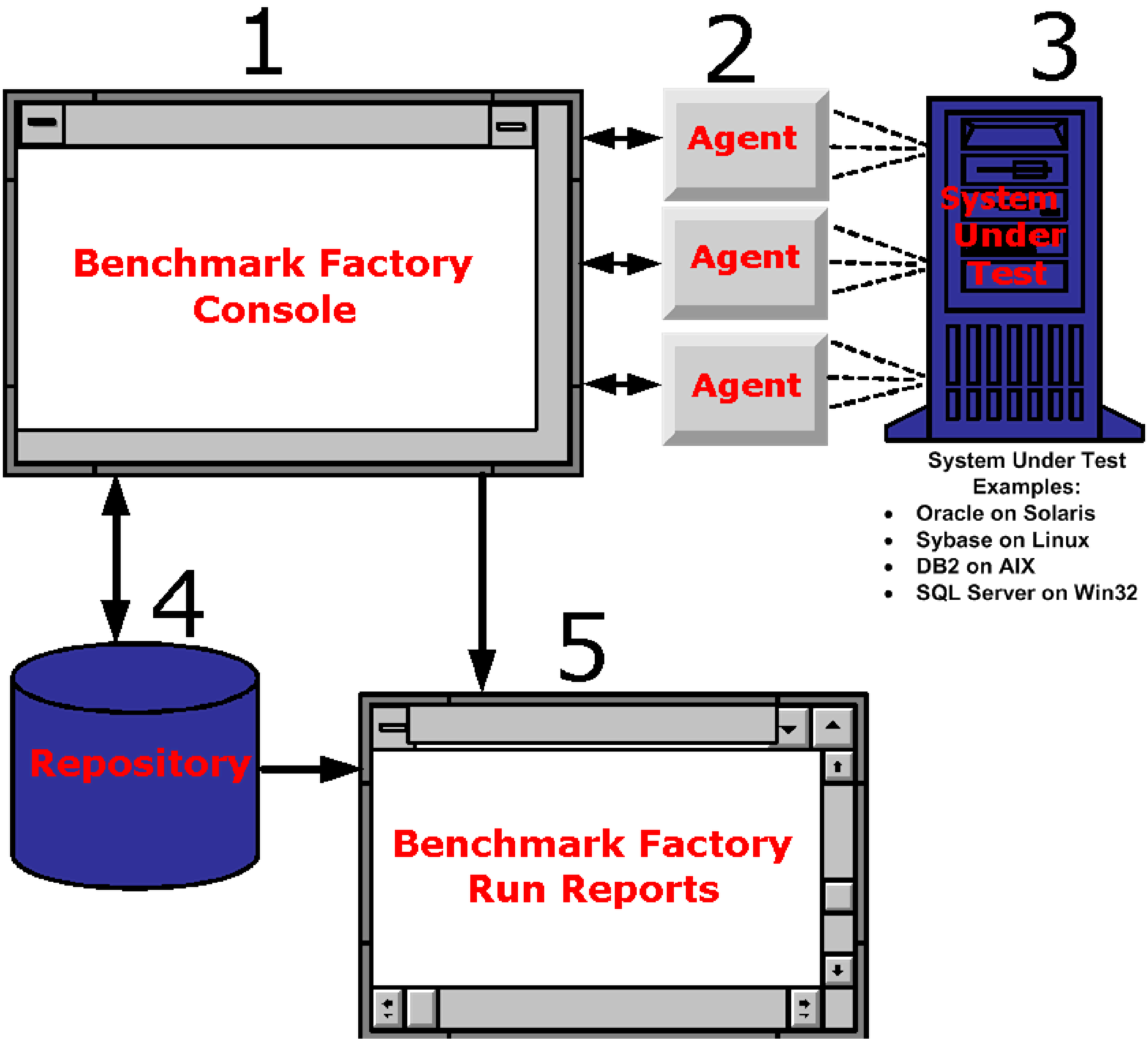
- The **Jobs View** pane is the user's workspace that allows you to create and save jobs.
    - Saved jobs are listed in the Jobs View pane. See Jobs View on page 59 for more information.
    - The **New Job Wizard** is the starting point for creating Jobs. See Quickstart: Create a New Job on page 60 for more information.
- A **job** is comprised of the following:
    - Job settings
    - A database connection
    - A workload (comprised of tests)
    - Create/Delete Benchmark Objects Test
    - An Execute test
- A **workload** is an assembled test comprised of user scenarios and/or transactions. These tests can be run with multiple virtual users. A workload or test can be of one of the following types:
    - **Mixed Workload**: A Mixed Workload test runs for a specified time at each predetermined user load level. Each user will run a transaction mix based upon the weights defined on the transactions tab. For example, if a test has two transactions, A and B, with A having a weight of one and B having a weight of four, on average B will run four times for every time A is run once. The run order will be randomly generated for each user so they are not all running the same transaction simultaneously. That run order is used for that user each time the test is performed to ensure reproducible results.
    - **Replay Test**: A Replay Test runs multiple transactions with each one running independently on a specified number of users. The test will run until the defined number of executions for each transaction or a specified time limit is reached.
    - **SQL Scalability Test**: A SQL Scalability test executes each transaction individually for each userload and timing or execution period. For example, a test could have two transactions, A and B, and two userloads of 10 and 20, with an iteration length of one minute. Transaction A would execute continually for one minute at userload 10, then B would do the same. Next A will run at userload 20, followed again by test B, for a total time of 4 minutes.
- **User Scenario:** A series of one or more transactions to be executed in order, to make a single transaction. A User Scenario is normally associated with user behavior simulated against the system-under-test. These are the components used to build a workload.
- **Transaction**: A single unit of work in the testing process, such as retrieving a Web page, executing a SQL statement, writing a file, or sending an email.
- **Execute Test:** Allows you to execute a file during the running of a job.
- **Create/Delete Benchmark Objects Test** configures a system-under-test for industry standard benchmark tests to measure system performance. Each standard benchmark has been developed with specific system configuration requirements, that include tables, indexes, data, etc.



How Benchmark Factory Works

The following steps provide an overview of how Benchmark Factory components interact during the load testing process.

1. The Benchmark Factory Console implements the workload testing process and controls one or more distributed agent machines. Each agent machine can simulate thousands of users. Each simulated user executes transactions and records statistics.
2. Benchmark Factory About Agents machines simulate virtual users. The Agents send transactions to the system-under-test. The Agents record statistics that includes how much data the resulting transaction contained and how long it took to get the results. At the end of an iteration, each agent machine reports its findings back to Benchmark Factory.
3. A server (system-under-test) is the database Benchmark Factory connects to. Benchmark Factory is server neutral and network protocol independent. Benchmark Factory uses vendor client libraries of the system-under-test. Any system that the client software can support, Benchmark Factory can support. Its only requirement is that Agent machines must be able to connect to a server through an appropriate manner and it supports the test system.
4. The Repository Manager stores all testing data.
5. Results are viewed from Run Reports.



Benchmark Factory Feature Comparison

Review the following table for a comparison of features available in the commercial and freeware editions of Benchmark Factory.

Feature	Freeware Edition	Commercial Edition
Default maximum virtual user load (without additional virtual users packs)	20	100
Additional virtual user packs	✗	✓ (unlimited)
SQL Optimizer Integration	✗	✓
Oracle Capture and Replay	✗	✓
SQL Server Capture and Replay	✗	✓

Feature	Freeware Edition	Commercial Edition
Maximum number of transactions that can be run during a load test/user scenario	20	Unlimited
Scaling factors for standard benchmarks (AS3AP, TPC-B, TPC-C, TPC-D, and Scalable Hardware)	Lowest two scaling factors for each benchmark	All are available
Repository database types	Only the default SQLite repository is enabled	The following repository database types are supported: Oracle, Microsoft SQL Server, SAP SQL Anywhere (Adaptive Server Anywhere), MySQL, and SQLite
BMFServer.exe	✗	✓
Support	Technical support from Quest is not available for the Freeware edition.	For the Commercial edition, support from Quest is available at: https://support.quest.com

✓ = Available

✗ = Not available

Get the latest product information and find helpful resources at the Benchmark Factory community at: http://www.toadworld.com/products/benchmark-factory

4

Benchmark Factory Components

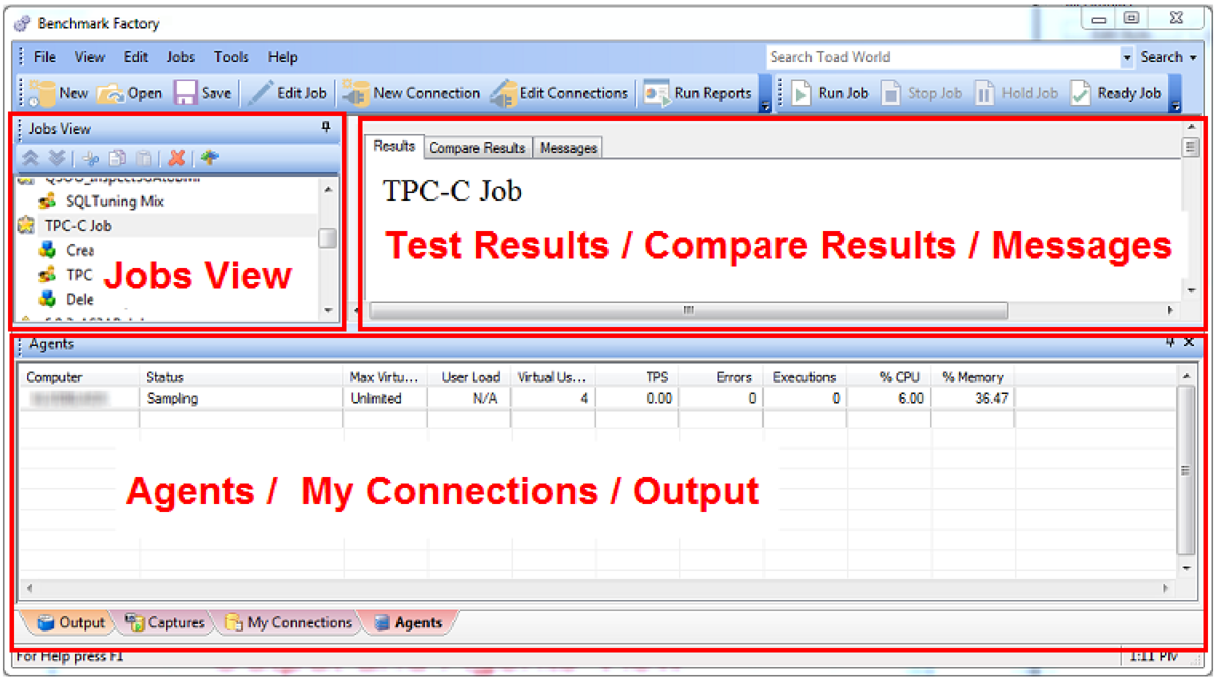
Benchmark Factory Console

The Benchmark Factory console implements the database workload testing process. This interface is where:

- Jobs are created and executed.
- Users can access test results. See Benchmark Factory Test Results on page 164 for more information.
- Agents are managed. See About Agents on page 36 for more information.

The Benchmark Factory Console contains the following views:

- Jobs View
- Benchmark Factory Test Results
- Agents View
- Output View



Computer	Status	Max Virtu...	User Load	Virtual Us...	TPS	Errors	Executions	% CPU	% Memory
[redacted]	Sampling	Unlimited	N/A	4	0.00	0	0	6.00	36.47

About Agents

Benchmark Factory Agents reside on each physical client machine and spawn multiple virtual-user sessions. Each Agent generates an information screen with tabs that contain a variety of Agent information. Benchmark

Factory can control hundreds of Agent machines.

Each Agent can simulate up to 1500 virtual users, depending upon hardware and test configuration. Virtual users simulate the load against the system-under-test. Each virtual user is a separate thread, acting independently of the other virtual users, with its own connection to the system-under-test. Each virtual user tracks its own statistics that includes transaction times and the number of times a transaction executes.

The Agents View (Pane)

The Agents view or pane in the main Benchmark Factory window displays the status of the agents currently running.

- If the Agents pane is not in view, select **View | Agent** from the main menu.

Add and Manage Agents

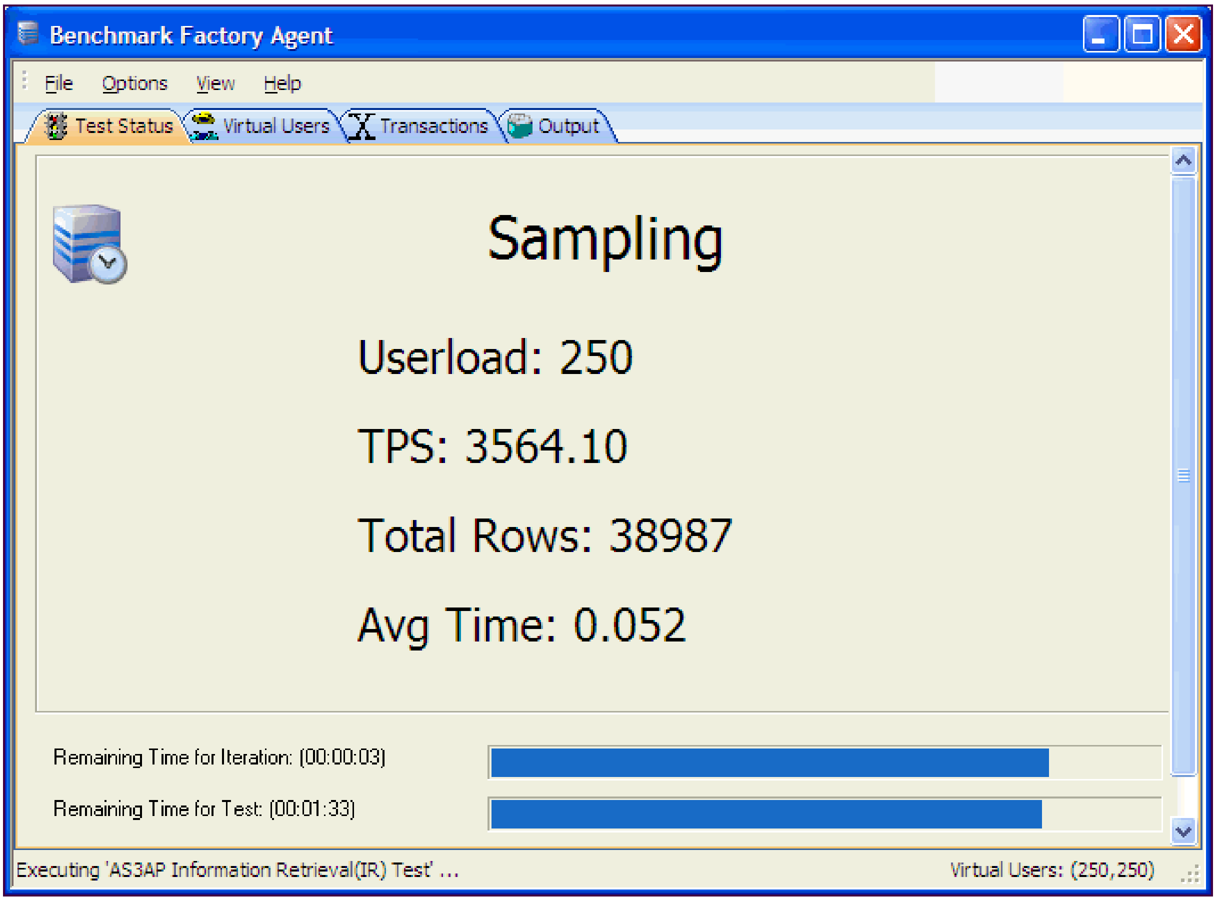
- To add agents or to specify default settings for agents, see Agent Settings .
- To install remote agents through the console, see Install Remote Agents.
- To learn about running Benchmark Factory with multiple agents, see Running Benchmark Factory with Multiple Agents.
- To learn about the non-UI Agent, see BMFAgent.exe.

The Benchmark Factory Agent Component Dialog

To open the Benchmark Factory Agent dialog, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe**.

The Benchmark Factory Agent dialog provides the follow tabs:

- Agent Test Status Tab
- Agent Virtual Users Tab
- Agent Transaction Tab
- Agent Output Tab



Sampling

Userload: 250

TPS: 3564.10

Total Rows: 38987

Avg Time: 0.052

Remaining Time for Iteration: (00:00:03)

Remaining Time for Test: (00:01:33)

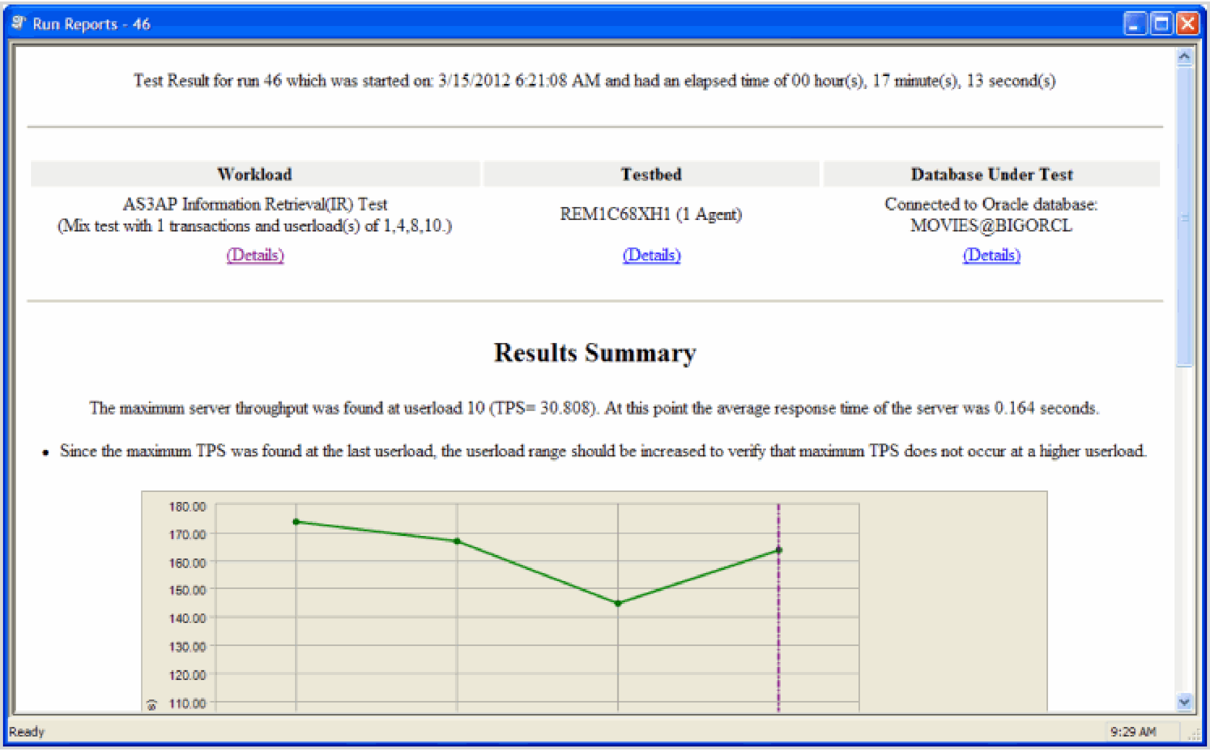
Executing 'AS3AP Information Retrieval(IR) Test' ...

Virtual Users: (250,250)

Run Reports

Benchmark Factory Run Reports is a separate executable that provides a comprehensive and detailed collection of database load testing results. With Benchmark Factory you can drill down into a database to view a wide array of information and statistics that gives you accurate insight into database performance. Run Reports Viewer allows you to access Benchmark Factory load testing results.

Note: Three instances of Run Reports can be viewed at one time.



Test Result for run 46 which was started on: 3/15/2012 6:21:08 AM and had an elapsed time of 00 hour(s), 17 minute(s), 13 second(s)

Workload	Testbed	Database Under Test
AS3AP Information Retrieval(IR) Test (Mix test with 1 transactions and userload(s) of 1,4,8,10.) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

Results Summary

The maximum server throughput was found at userload 10 (TPS= 30.808). At this point the average response time of the server was 0.164 seconds.

- Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

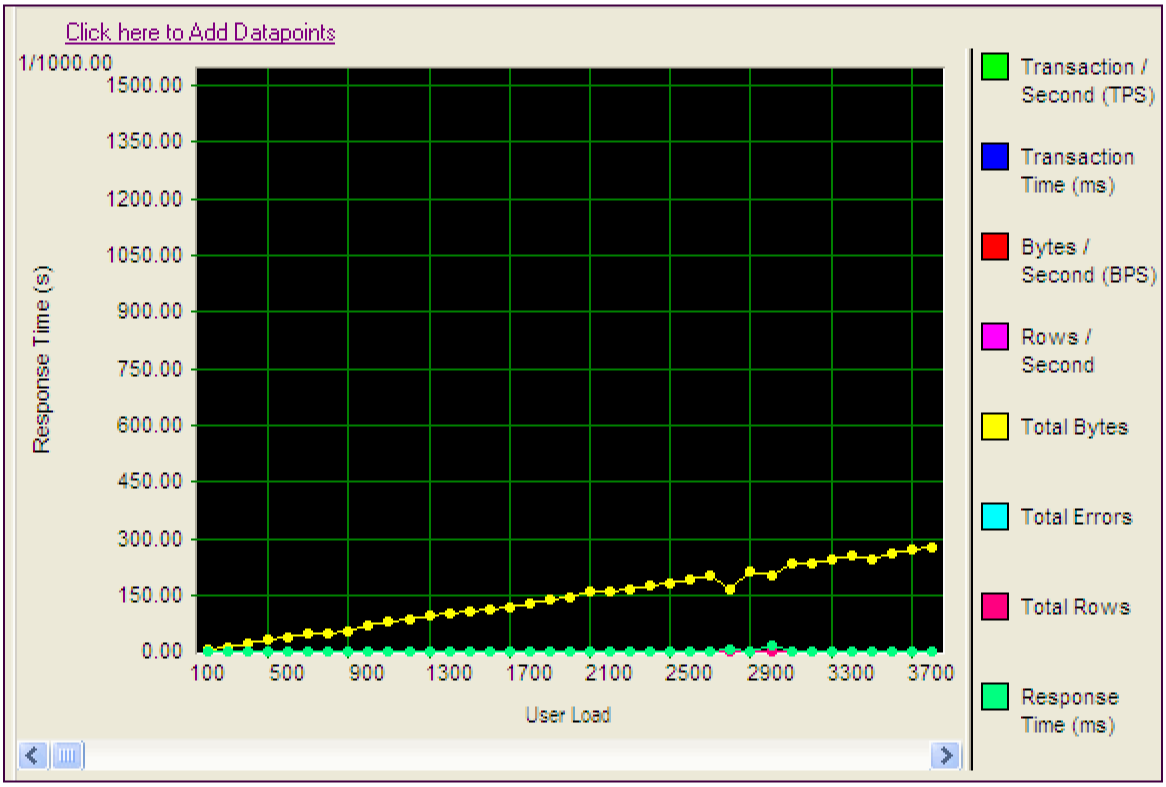
Cutting and Pasting of Run Reports

From Benchmark Factory run reports you can cut and paste test results into the Clipboard.

Testing Server Throughput

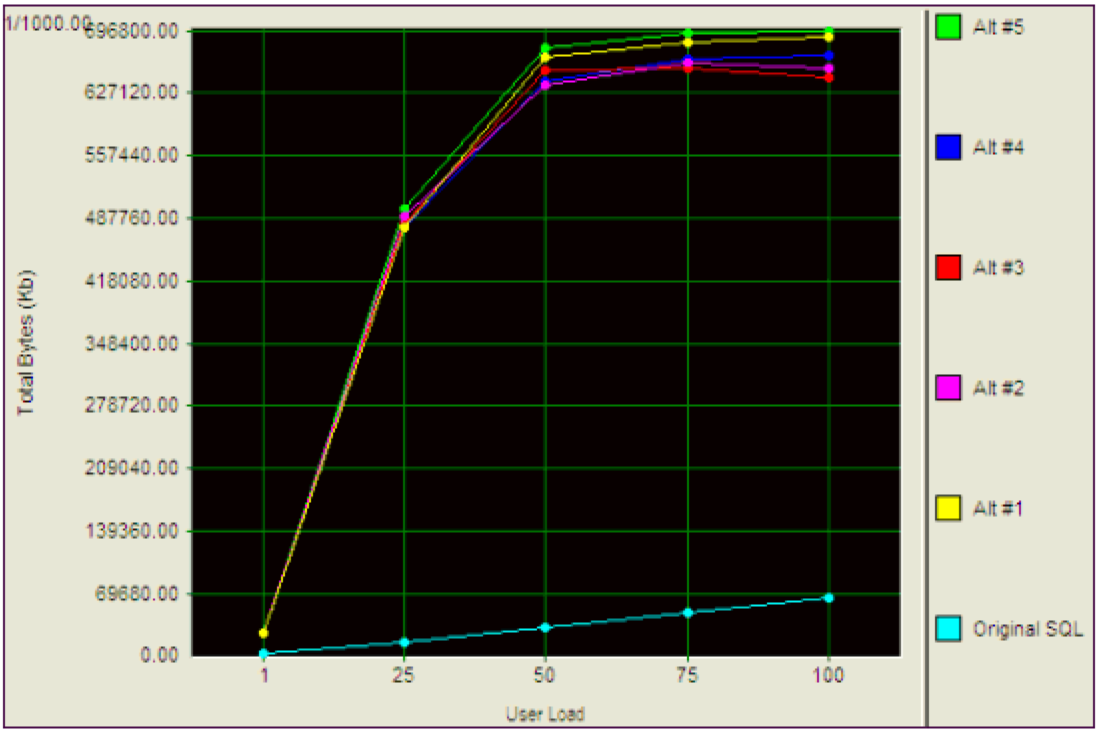
Server throughput is the measurement of database capacity. Benchmark Factory allows you to view a variety of server throughput statistics that include:

- Transaction/Second (TPS)
- Transaction Time
- Bytes/Second (BPS)
- Rows/Second (RPS)
- Total Bytes
- Total Errors
- Total Rows
- Response Time



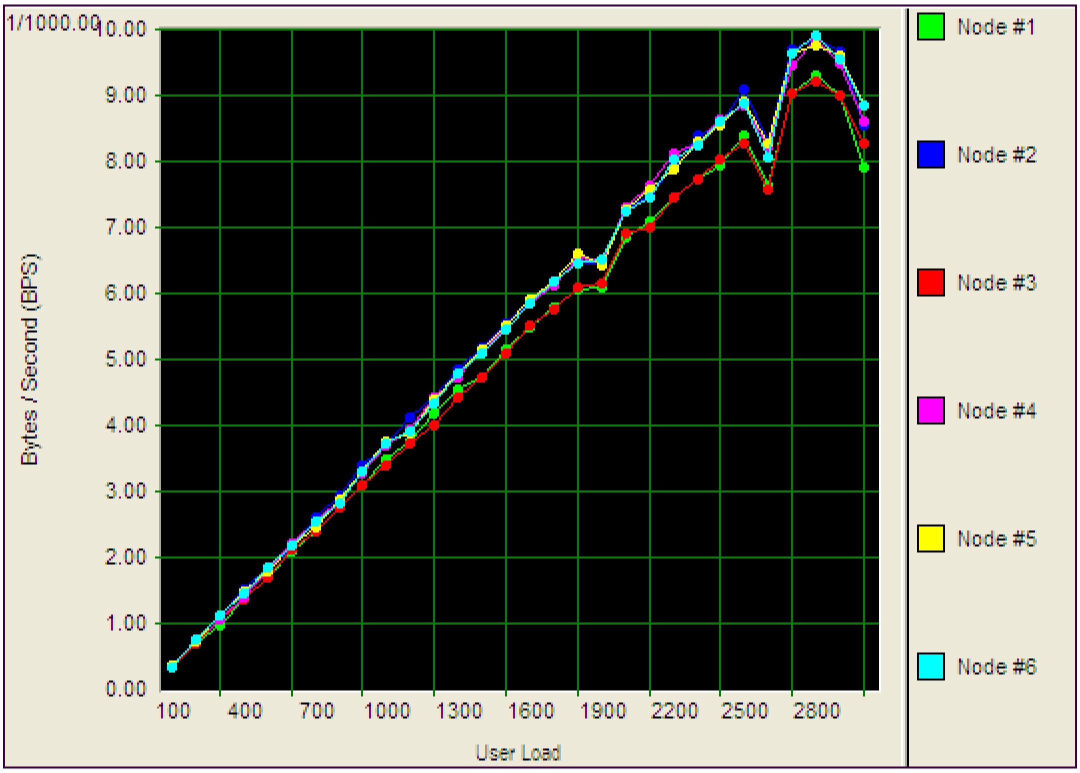
Testing SQL Scalability

The Benchmark Factory transaction comparison graph provides users with a comparison of the individual transaction performance as they compare to each other. This is most useful when analyzing a SQL Scalability test.



Testing Real Application Clusters

Benchmark Factory allows you to load test Oracle Real Application Clusters (RAC) to determine system capacity. The cluster performance graph allows users to view the performance of individual nodes.



Repository Manager

Note: If you create a new Benchmark Factory 5.5 (or later) repository, earlier versions of Benchmark Factory will not work against this repository.

The Repository is a database where all of the test results are stored. Benchmark Factory inserts test results into the repository and provides an easy way to access the data. By default, the Repository is a SQLite database that resides on the same machine as Benchmark Factory. The Repository can reside on another database server if required.

Note: By default in Benchmark Factory 7.1.1 or earlier, a MySQL database is created and used as the Repository, unless you selected the SQLite option during installation. In Benchmark Factory 7.2 or later, by default a SQLite database is created and used as the Repository.

To change the database, select the Data Source Name of the ODBC connection for the new database. To migrate data from one database to another, click **Data Migration** to open the Data Migration Wizard.

Note: If the database structure does not exist on the selected database, a prompt to create the structure will appear when **OK** is clicked.

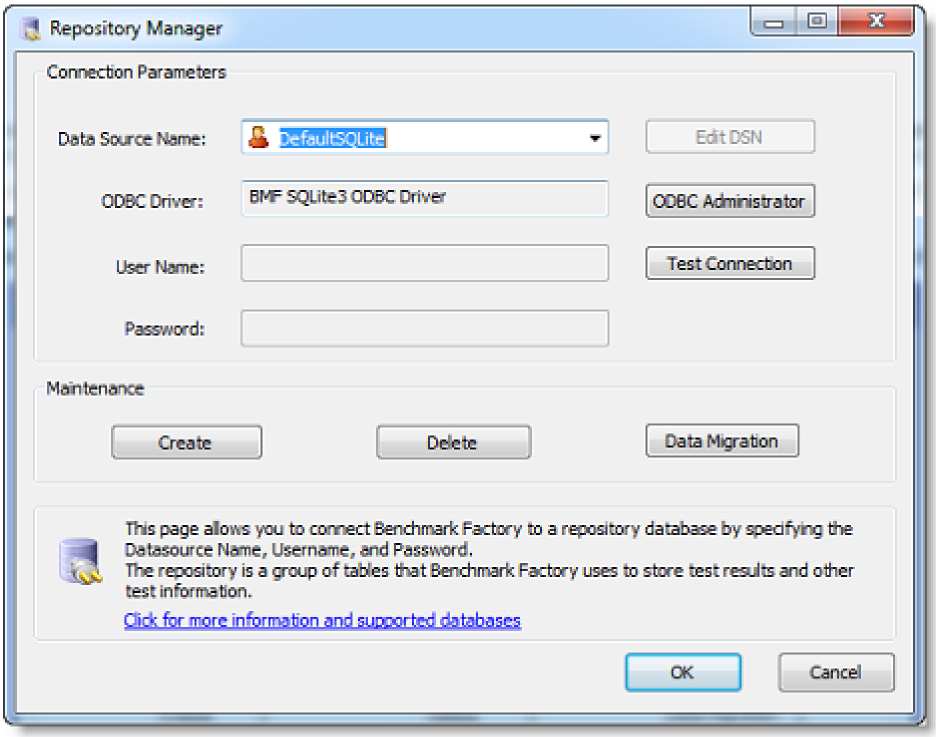
The supported database types are:

- MySQL
- SQL Server®
- SQLite

- SAP SQL Anywhere® (Adaptive Server® Anywhere)
- Oracle®

To open the Repository Manager

1. To open the Repository Manager from the application, select **Tools | Repository Manager**. Benchmark Factory prompts you to close the application. This is necessary before opening the Repository Manager.
2. The Repository Manager opens. Modify connection parameters or perform maintenance functions.



Repository Manager

Connection Parameters

Data Source Name: DefaultSQLite

ODBC Driver: BMF SQLite3 ODBC Driver

User Name:

Password:

This page allows you to connect Benchmark Factory to a repository database by specifying the Datasource Name, Username, and Password.
The repository is a group of tables that Benchmark Factory uses to store test results and other test information.
Click for more information and supported databases

Review the following for additional information:

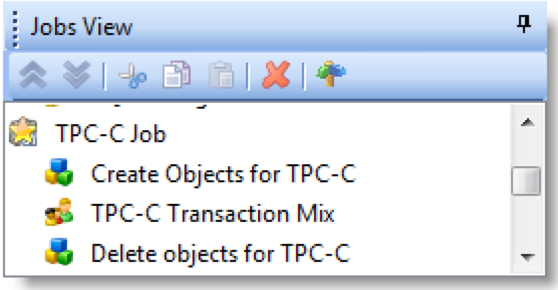
Connection Parameters	
Data Source Name	Data Source name of the ODBC connection used to connect to the repository database.
ODBC Driver	Current ODBC driver
User Name	The User Name used to log into the selected database.
Password	The Password associated with the user name used to log into the database.
Edit DSN	Displays the ODBC connection information dialog for the selected data source.
ODBC Administrator	Displays the ODBC Data Source Administrator dialog. Use this to add and edit ODBC connections.
Test Connection	Tests the connection of the currently selected ODBC Data Source.
Maintenance	
Create	Creates the repository objects on the selected database.
Delete	Deletes the repository objects on the selected database.

	ℹ **Warning**: This will delete all test results stored in the Repository.
Data Migration	Launches the Data Migration Wizard.

Jobs View

The Jobs View displays a list of saved jobs, shows which jobs are currently running, and which jobs are scheduled to run.

From the Jobs View pane, you can run jobs, view job status, and view real-time testing results.



To run a job

- Select the job in the Jobs View pane and click Run Job.

To view test results

- To view a job's test results, select a job in the Jobs View pane. Test results display in the right pane. See Benchmark Factory Console for an overview of the Benchmark Factory console.
- To compare two or more run results for a test, select the **Compare Results** tab. Use **Ctrl+click** to select multiple test runs. A comparison of the results for the various runs displays.

To edit an existing job

- To edit an existing job (to modify the workload or job setup options), select the job in the Jobs View pane and click Edit Job.

View Job Status

From the Jobs View pane, you can view job status.

The following job states are identified:

- **Scheduled**: All jobs currently waiting to run or scheduled to run at a future time.
- **Running**: Job currently running.
- **Completed**: All completed jobs.

Jobs View Toolbar

The Jobs View toolbar provides user functionality to save or delete jobs.

⇈	Move the selected test up in the list.
⇊	Move the selected test down in the list.
✗	Delete the selected item from the Jobs View.
🌳	Toggles tree-view on and off.

Note: To save a job as a Benchmark Factory script, select the job and click **Save** in the Benchmark Factory toolbar or select **File | Save**.

BMFServer.exe

BMFServer.exe is a non-UI executable installed with Benchmark Factory. BMFServer.exe performs the same functionality as Benchmark Factory, except BMFServer.exe has no graphic user interface. This allows you to easily integrate BMFServer.exe into your continuous integration or continuous testing process.

Note: This feature is not available in the freeware edition of Benchmark Factory.

Details

- BMFServer.exe is installed into the **bin** sub-directory of the installation directory at the time you install Benchmark Factory.

  The default location is: C:\Program Files\Quest Software\Benchmark Factory <version>\bin

- You cannot run the Benchmark Factory console and BMFServer.exe at the same time on the same machine.

- To enter or modify your Benchmark Factory license, you must use the Benchmark Factory console, not BMFServer.exe.

How to Use BMFServer.exe

Use BMFServer.exe when automating your continuous improvement / continuous testing process. You can run BMFServer.exe using a script, a custom application, or the Command Prompt window. The REST API included with Benchmark Factory can be used with BMFServer.exe.

- To modify settings, use the BMFServer.ini file located in the same directory as BMFServer.exe or use the Command Prompt window.

- The default REST API port number for BMFServer.exe is port 30110. (The default port number for the Benchmark Factory console is port 30100.)

- The default TCP/IP port for BMFServer.exe is the same as for the Benchmark Factory console, port 4568.

Start BMFServer.exe

To Start BMFServer.exe

1. Open the installation directory. The default installation path is

   C:\Program Files\Quest Software\Benchmark Factory <version>

2. Open the **bin** directory.
3. Double-click **BMFServer.exe**.

   ℹ **Note:** You cannot run BMFServer.exe and BFactory.exe simultaneously.

5

Getting Started - the Benchmark Factory Workflow

This topic outlines the Benchmark Factory workflow. Click the links in each step to drill down to more-detailed information.

Step 1. Set Up Benchmark Factory

Use the Settings dialog to specify default settings for job and workload options, such as benchmark scale, the number of virtual users per iteration, timing, and latency. Changes to these settings affect only new jobs, not existing jobs.

To specify Benchmark Factory settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select a page from the left-side navigation and then specify the corresponding options. See About Settings on page 151 for more information.

Step 2. Create a Connection

Before you can run a job against a database, the database connection must be defined in Benchmark Factory. You can define a new connection from within the New Job Wizard, or you can define the connection using the New Connection dialog or the My Connections tab prior to creating a new job.

To create and save a new connection

1. Click in the main toolbar. The New Connection dialog opens.
2. Select a connection type from the drop-down list. The Connection properties for that connection type display.
3. Enter the connection information and click **Create Connection**. For more information about creating connections, see About Benchmark Factory Connections.

Step 3. Create a New Job

Use the New Job Wizard to create a new job. The New Job Wizard guides you through the steps required to create a database load test (workload) and specify general options for the job.

When creating a new job in the wizard, you can specify settings that apply to this job and override the defaults in the Settings dialog. See Quickstart: Create a New Job on page 60 for more information.

To create a new job

1. Select **New** from the Benchmark Factory main toolbar. The New Job Wizard opens.
2. Select the connection for the database-under-test. See About Benchmark Factory Connections on page 128 for more information.
3. Click **Workload** in the left pane to begin specifying a workload.
    a. On the Workload page, select the type of test from the drop-down list (for example, Industry Standard Benchmark). Then select a test to include in the workload.
    b. After selecting a test, click the **Select Test** or **Add Test** button at the bottom of the wizard page to specify options for that test.
4. To add another test to the workload, click **Add Another Test/Step**.
5. When you finish specifying the workload for the job, click **Agent** in the left pane to specify an agent.
    - Click the checkbox to the left of the agent name to select it. A checkmark displays for the selected agent.
6. Click **Save/Close** to save the job and close the wizard.
7. The new job is added to the list of jobs in the Jobs View pane.
8. To rename the job, right-click the job in the Jobs View and select **Rename**.

Modify a Job

After creating a new job, you can modify the job. For example, you can change the database-under-test, add tests/steps, or change test options.

To modify an existing job

1. In the Jobs View pane, right-click the job and select **Edit Job**.
2. To modify the number of virtual users for a specific test in a job, do the following:
    a. Right-click the test in the Jobs View pane and select **Edit Job**. The Edit Job wizard opens to the selected test in the workload.
    b. Select the Test Options tab. Then select the User Load tab.
    c. Modify the User Load. Click **Save/Close** when finished.

Step 4. Run the Job

To run a job

- In the Jobs View, select the job to run and click Run Job in the jobs toolbar, or right-click the job and select **Run Job**.

Step 5. Analyze Test Results

Benchmark Factory provides a number of metrics to analyze a system-under-test. These include:

- Response, retrieval, and transactions times
- Transaction throughput in transactions per second (TPS)

- Amounts of data transferred in bites per second (BPS)
- Errors incurred
- Transaction Distribution
- Any user defined counters (Windows NT/2000 Performance Monitor) added using the Benchmark Factory Console.

Benchmark Factory provides the following testing results:

- Real time statistics (See View Test Results - Running Job on page 164 for more information.)
- View results and compare results for completed tests (See View Test Results - Completed Job on page 172 for more information.)
- Run Reports (See Run Reports on page 173 for more information.)
- Testing results exported to Excel (See Export Benchmark Factory Test Results to Excel on page 195 for more information.)

6

Agents

About Agents

Benchmark Factory Agents reside on each physical client machine and spawn multiple virtual-user sessions. Each Agent generates an information screen with tabs that contain a variety of Agent information. Benchmark Factory can control hundreds of Agent machines.

Each Agent can simulate up to 1500 virtual users, depending upon hardware and test configuration. Virtual users simulate the load against the system-under-test. Each virtual user is a separate thread, acting independently of the other virtual users, with its own connection to the system-under-test. Each virtual user tracks its own statistics that includes transaction times and the number of times a transaction executes.

The Agents View (Pane)

The Agents view or pane in the main Benchmark Factory window displays the status of the agents currently running.

- If the Agents pane is not in view, select **View | Agent** from the main menu.

Add and Manage Agents

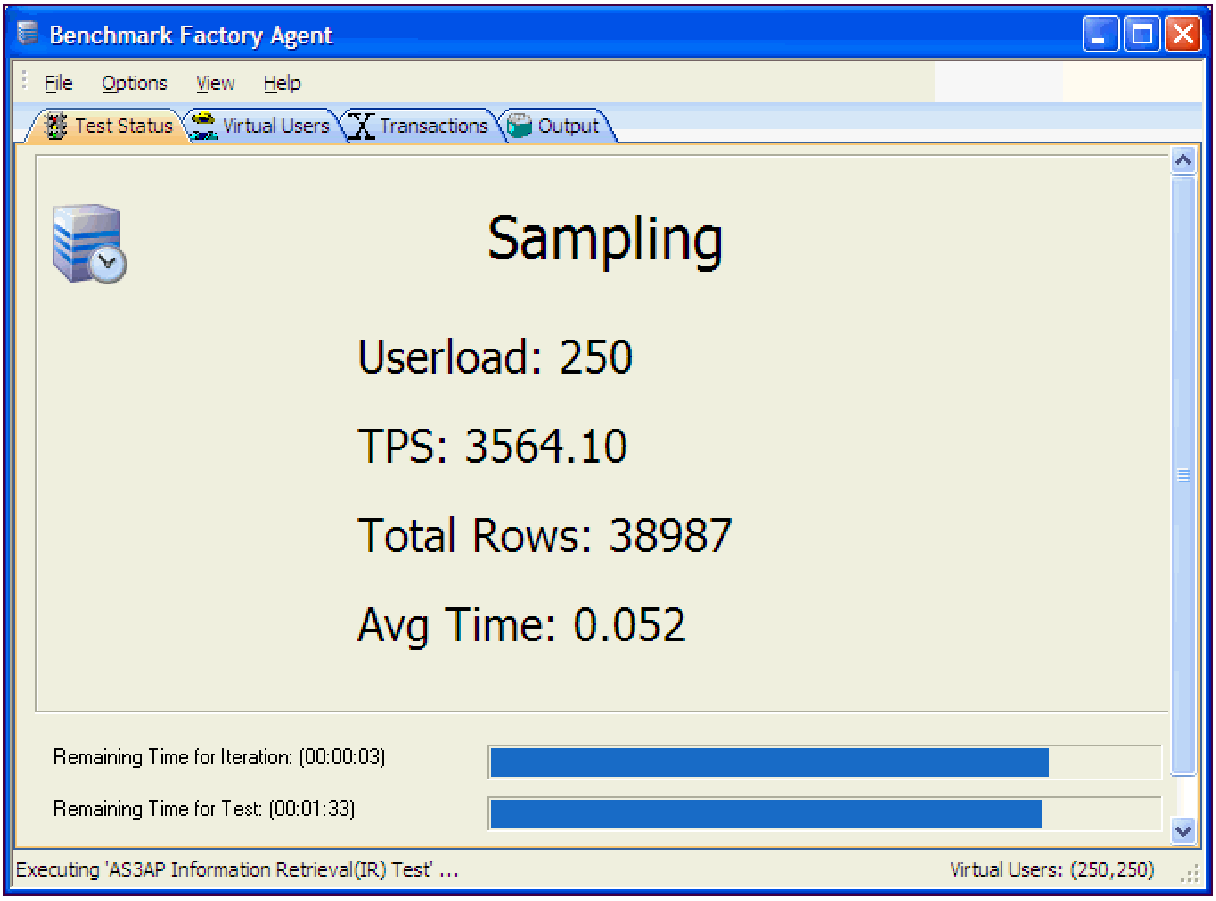
- To add agents or to specify default settings for agents, see Agent Settings .
- To install remote agents through the console, see Install Remote Agents.
- To learn about running Benchmark Factory with multiple agents, see Running Benchmark Factory with Multiple Agents.
- To learn about the non-UI Agent, see BMFAgent.exe.

The Benchmark Factory Agent Component Dialog

To open the Benchmark Factory Agent dialog, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe**.

The Benchmark Factory Agent dialog provides the follow tabs:

- Agent Test Status Tab
- Agent Virtual Users Tab
- Agent Transaction Tab
- Agent Output Tab



Sampling

Userload: 250

TPS: 3564.10

Total Rows: 38987

Avg Time: 0.052

Remaining Time for Iteration: (00:00:03)

Remaining Time for Test: (00:01:33)

Executing 'AS3AP Information Retrieval(IR) Test' ...

Virtual Users: (250,250)

Agent Settings

You can use this page of the Settings dialog to:

- Specify some default settings (error handling and communication timeout) for each agent
- Set up (add) new agents to use in testing
- Specify the defaults agents to use when creating a new job.
- Install remote agents

To specify default Agent settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select **Agent**. Review the following for additional information:

Setup New User Agent	Click to setup a new agent.
List of agents	Specify default settings. Select the agent machines and specify the number of agents. These default settings are used when you create a new job. If you do not select a default agent, the local agent is automatically selected when you create a new job in the New Job wizard.

	Note: When you run a job using one or more local agents, if Agent utilization of resources on the local machine is too high, errors could occur.
Remove	Click to remove the selected agent.
Verify Selected Agent	Verifies that selected agent starts and communicates with console.
Error Handling	**Stop job when agent cannot be started/connected**—Cancels job if agent does not start or connection is lost. Do not select this option if you want to run a job using multiple agents AND you want the job to run even if one or more agents do not start.
Agent communication timeout	Specify timeout duration in seconds.
Agent refresh interval	Specify how often each agent should refresh its status.

To set up or to install an Agent

1. Select **Edit | Settings | Agent**.
2. On the Agent settings page, click **Setup New User Agent**. The Setup User Agent dialog opens.
3. Enter the name or IP address of the computer where the agent is installed, or the computer on which you want to install an agent.
   - To install an agent on a remote computer, see Install Remote Agents.
4. Enter the connection information.
5. Specify the number of agents. Click **OK**.
6. Benchmark Factory connects to the computer and then checks if the agent is installed.
7. The agent is added to the list.
   - To learn more about running multiple agents, see Running Benchmark Factory with Multiple Agents.

Tips:

- You can also modify agents or set up new agents for a job using the New/Edit Job Wizard. Select **Agent** in the left pane of the wizard to access agent options for the selected job.
- To open the Agent console, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe**. See About Agents on page 36 for more information.

Install Remote Agents

You can install agents to remote computers from your Benchmark Factory console. You must be able to connect to the remote machine.

To install remote agents from the console

1. Select **Edit | Settings | Agent**.
2. On the Agent Settings page, click **Setup New User Agent**. The Setup User Agent dialog opens.
3. Enter the connection information for the remote machine. Review the following for additional information:

IP Address / Name	Enter the IP address or name of the remote machine.
User Name	Enter the user name to use to connect. ℹ **Note:** Only accounts in the Administrators group on the remote machine can install or launch remote Agents. If in a domain, connect using a domain account in the Administrators group.
Password	Enter the password to use to connect.
Use Windows session credentials	Select to use Windows credentials.
# of Agents	Specify the number of agents to deploy on this machine.

4. Click **OK**. Benchmark Factory checks the remote machine for an installed agent.
5. If no agent is found, Benchmark Factory prompts you to install the agent. Click **Yes** in the prompt window. The Setup User Agent dialog opens.
6. In the **Installer** field, browse to and select the Benchmark Factory installer.

   ℹ **Note:** The Installer can be located on your local machine or on the remote machine.
7. Click **OK**. The installer installs the agent component on the remote machine.

   ℹ **Note:** If the agent fails to install, you may need to configure the firewall or attempt one of the troubleshooting techniques. See Configure Firewall for Remote Agent Install/Start-Up.
8. When the agent is successfully installed, the progress window closes and the new remote agent displays in the list.
9. After the remote agent is installed, you can double-click the agent name in the list to modify agent options.

Configure Firewall

In order to install remote agents and allow communication with remote agents after installation, you may need to configure the firewall on the console machine and on each agent machine. See Configure Firewall for Remote Agent Install/Start-Up to learn more.

If you configure the firewall and then encounter an error when attempting to install a remote agent, find some troubleshooting techniques here: Troubleshooting.

Running Benchmark Factory with Multiple Agents

You can install agents on multiple agent machines in order to run tests using multiple agents. To configure Benchmark Factory with multiple agents you must:

- Load the proper client libraries (i.e. Oracle, SQL server, etc.) on the Benchmark Factory Console and Agent Machine.
- Install Benchmark Factory on the local and agent machines.

Install Multiple Agents

For best results, install remote agents through your Benchmark Factory console. You can install remote agents from the Settings dialog (Edit | Settings | Agent) or from the Agent page of the New/Edit Job Wizard. See Install Remote Agents on page 38 for more information.

You can also install agents manually on the agent machine. If you do, you must configure the agent on the agent machine (see below). Then, in your local Benchmark Factory, you must set up the agent to register it with the console. See Agent Settings on page 160 for more information.

Configure Agent Machine

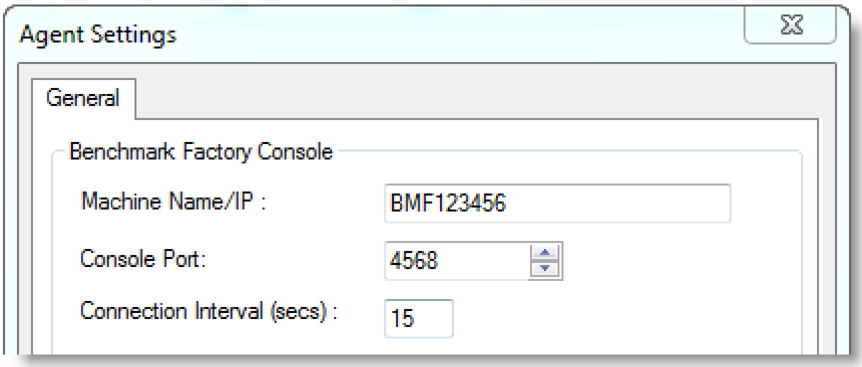
If you set up an agent machine by installing Benchmark Factory manually, you must configure the agent on the agent machine.

Each Benchmark Factory Agent must be configured with the address of the Benchmark Factory Console. Each Agent sends load testing results back to the Benchmark Factory Console.

If you use only the agent installed locally on the console machine, make sure your local agent is configured with the IP address (name) of your local machine.

To configure Benchmark Factory Agents

1. On each agent machine, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe** to open the Agent dialog.
2. Select **Options | Settings**.
3. In **Machine Name/IP** field, enter the address of the Benchmark Factory console machine. In the **Console Port** field, enter the port for the Benchmark Factory console.



Machine Name/IP : BMF123456

Console Port: 4568

Connection Interval (secs) : 15

4. Click **OK**.
5. Repeat this procedure for each Agent machine

Set Up the Agents

To run a job using multiple agent machines, after the agents are installed and configured, you must set up (or add) each remote agent to the Benchmark Factory console.

- If the agent was installed manually, you can set up the agent in **Edit | Settings | Agent**. See Agent Settings on page 160 for more information.
- If the agent was installed remotely through the Benchmark Factory console, the agent is automatically set up.

Running a Job With Multiple Agents

To run a job with multiple agents

1. Open the New Job Wizard and create a workload.
2. Click **Agent** in the left panel of the New Job Wizard.
   - For an existing job, open the job in the Edit Job Wizard and click **Agent** in the left pane.
3. From the list, select the agents to use. The list displays agents you have previously set up in Benchmark Factory.
   - To add an agent, click **Setup New User Agent**. See Agent Settings on page 160 for more information.
4. As the job runs, all connected Agents will display in the Agent view/pane of the Benchmark Factory console.

Note: When you run a job using one or more local agents, if Agent utilization of resources on the local machine is too high, errors could occur.

View Agents from Benchmark Factory Console

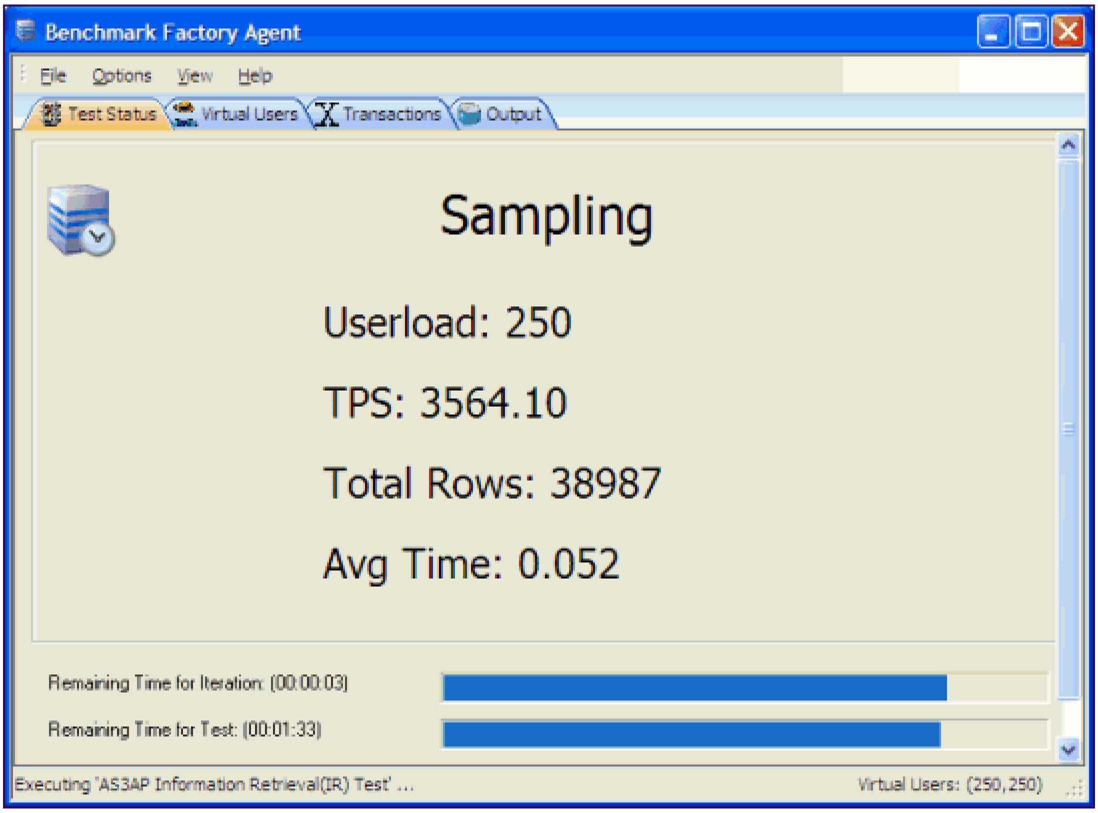
When you run multiple Agents for load testing, you can view the Agents results as they run from the Benchmark Factory Console.

To view the Agents from the Benchmark Factory Console

1. From the Benchmark Factory main menu, select **View | Agent**. The Agent view displays. Statistics from All Agents connected to the Benchmark Factory Console and running are displayed here.

Agent Test Status Tab

The Test Status tab displays information about the test you are currently running.



Sampling

Userload: 250

TPS: 3564.10

Total Rows: 38987

Avg Time: 0.052

Remaining Time for Iteration: (00:00:03)

Remaining Time for Test: (00:01:33)

Executing 'AS3AP Information Retrieval(IR) Test' ...

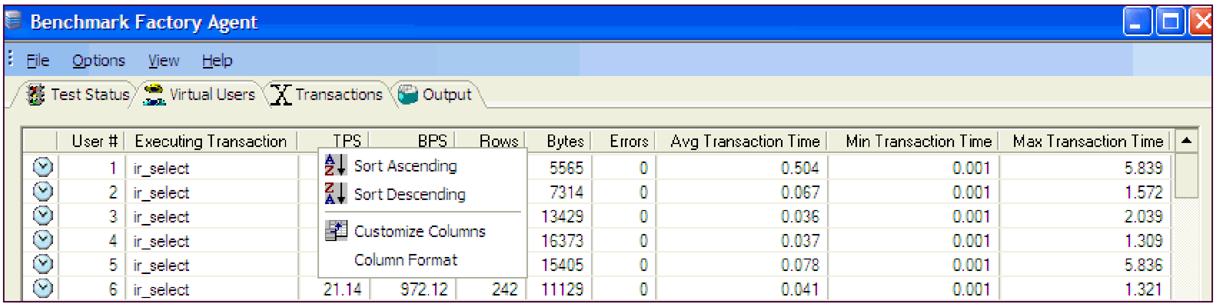
Virtual Users: (250,250)

Agent Virtual Users Tab

The Virtual Users tab provides the raw data from each virtual user. A tabular format shows each virtual user with its measurements.

To adjust or customize Virtual Users columns

1. Right-click on the Virtual Users column heading. A drop-down displays.

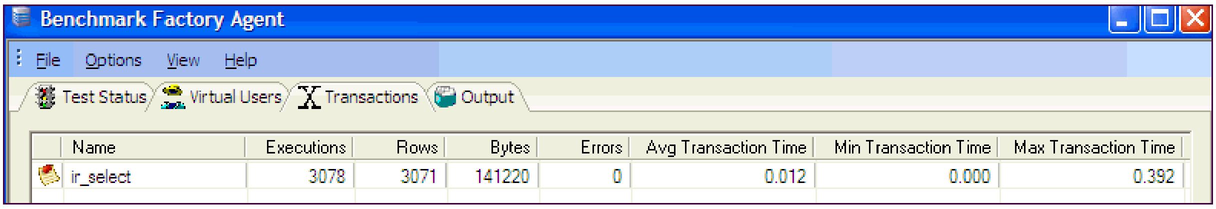


	User #	Executing Transaction	TPS	BPS	Rows	Bytes	Errors	Avg Transaction Time	Min Transaction Time	Max Transaction Time
	1	ir_select				5565	0	0.504	0.001	5.839
	2	ir_select				7314	0	0.067	0.001	1.572
	3	ir_select				13429	0	0.036	0.001	2.039
	4	ir_select				16373	0	0.037	0.001	1.309
	5	ir_select				15405	0	0.078	0.001	5.836
	6	ir_select	21.14	972.12	242	11129	0	0.041	0.001	1.321

2. Make the desired adjustments.

Agent Transaction Tab

The tab displays status information, including average, minimum, and maximum times on all active transactions.



Name	Executions	Rows	Bytes	Errors	Avg Transaction Time	Min Transaction Time	Max Transaction Time
ir_select	3078	3071	141220	0	0.012	0.000	0.392

Agent Output Tab

The Output tab is the same as the Messages window. This tab displays messages, debug, and results information.

BMFAgent.exe

BMFAgent.exe is a non-UI agent included with Benchmark Factory. BMFAgent.exe performs the same functionality as Agent.exe, except BMFAgent.exe has no graphic user interface. This allows you to easily integrate BMFAgent.exe into your continuous integration or continuous testing process.

Details

BMFAgent.exe is installed into the **bin** sub-directory of the installation directory at the time you install Benchmark Factory.

BMFAgent.exe has a smaller footprint (file size) than Agent.exe.

How Benchmark Factory Uses BMFAgent.exe

Because BMFAgent.exe has the same functionality as Agent.exe, Benchmark Factory uses BMFAgent just as it would Agent.exe. Review the following for additional information.

- On your local machine, the Benchmark Factory console attempts to use Agent.exe first. If Agent.exe is not found, the console uses BMFAgent.exe.
- When a job uses a remote agent, if Benchmark Factory cannot find BMFAgent.exe on the remote machine, Agent.exe is used on the remote machine.
- You can run multiple instances of BMFAgent.exe at the same time on the same machine.

Start BMFAgent.exe

To Start BMFAgent.exe

1. Open the installation directory. The default installation path is

   C:\Program Files\Quest Software\Benchmark Factory <version>

2. Open the **bin** directory.
3. Double-click **BMFAgent.exe**.

Modify Settings

When BMFAgent.exe and Agent.exe are installed, default settings are applied, such as the machine name (IP address) and the port number for the Benchmark Factory console to which the agent connects.

Use one of the following methods to modify the BMFAgent.exe settings:

- Use the **BMFAgent.ini** file located here: C:\ProgramData\Quest Software\BMF.
- Use the Command Prompt window.
- Open the Agent Settings dialog through the Benchmark Factory console.
  1. Start BMFAgent.exe.
  2. In the console, select **View | Agent** to open the Agents tab.
  3. Right-click the BMFAgent and select **Settings**.

Command Line Parameters

The following parameters are available from the command line.

Parameter	Description
/p <*port number*>	Changes the BMFAgent port used to connect to the console Example: BMFAgent.exe /p 2345
/c <*console name/ip*>	Changes the console machine name/ip to which the Agent connects Example: BMFAgent.exe /c *myservername*
/s	Automatically shuts down the Agent when the job is finished Example: BMFAgent.exe /s

7

Connections

About Benchmark Factory Connections

Benchmark Factory allows you to create the following connections:

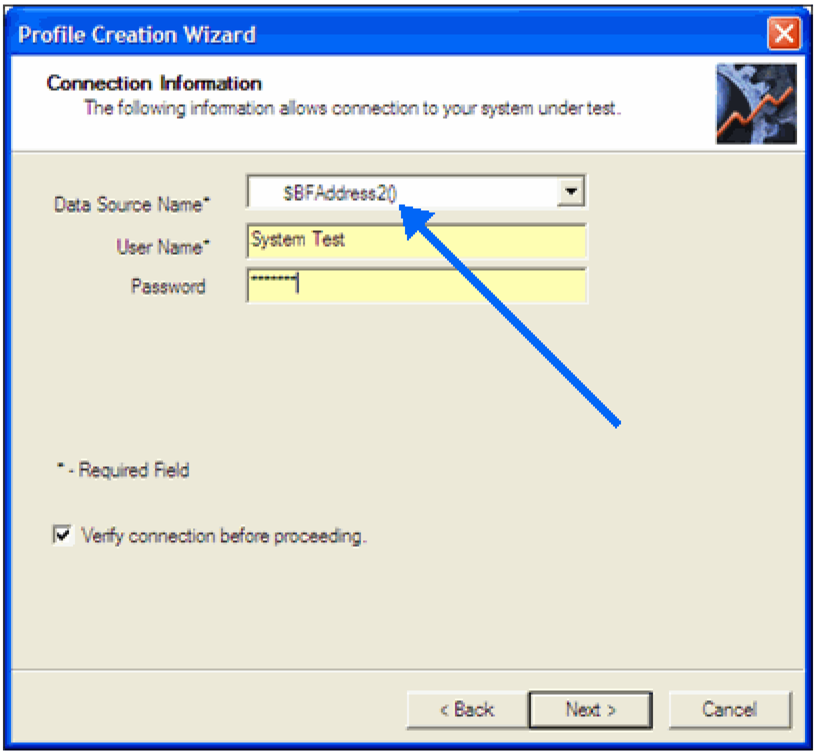
- IBM DB2CLI
- Microsoft SQL Server
- MySQL (ODBC)
- ODBC
- Oracle
- SAP

Adding BFScript to the Connection Dialog

You can add BFScripts to the Benchmark Factory Data Source Names, allowing you to have individual agents test against a different server. BFScripts have been enabled on the User name and Password fields.

To add BFScripts to the Data Source Name

1. Click inside **Data Source Name**.
2. Type the desired script name.



Connection Information

The following information allows connection to your system under test.

Data Source Name* $BFAddress2()

User Name* System Test

Password ********

* - Required Field

☑ Verify connection before proceeding.

Create Oracle Connection

Benchmark Factory allows you to create an Oracle connection using an Oracle client, direct connecton, or ODBC.

To create an Oracle connection

1. Click New Connection in the main toolbar. The New Connection dialog opens.
2. Select **Oracle** from the drop-down list.
3. Enter the connection information. Review the following for additional information:

   Note: BFScripts have been enabled on the User Name and Password fields.

Native tab	
User/Schema	Schema to which you want connect.
Password	Password for schema to which you want connect.
TNS or Direct tab	TNS—Allows you to connect to a database using your TNS names file. • Databases—Allows you to connect to a database using your TNS names file. Select a database from the list. Direct—Allows you to connect to a database using Host, Port, Server name or SID.
Connect as	Type of connection you connect to the database with: Normal,

	SYSDBA, or SYSOPER.
Connect Using	Select the Oracle Client to use for this connection.
Make this the BMF default home	Selecting this checkbox sets this as the default client for Benchmark Factory.
Connection Name	Enter a name to use to identify the connection in the My Connections pane.
ODBC tab	
Data Source Name	Select a data source from the drop-down list. Click Add DSN to create a new data source.
User Name	Enter the user name to use for this connection.
Password	Enter the password to use for this connection.
Reconnect Options	Click link to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

4. Click **Create Connection** to save the connection information and connect.
5. After creating a new Oracle connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

Miscellaneous Options

Select the **Miscellaneous** tab of the Connection dialog to specify options for flushing cached data. Review the following for additional information.

Note: The Miscellaneous tab is only available when creating an Oracle Native connection. For an Oracle ODBC connection, the default options specified in **Edit | Settings | Oracle** are used.

Database Flush	
Flush data buffer caches at start of each test iteration	Select to clear data buffer caches between iterations. ℹ **Note:** To perform this action, the Oracle database account must have certain privileges. In Oracle 9i, the ALTER SESSION privilege is required. In Oracle 10g or later, the ALTER SYSTEM privilege is required.
Flush shared pool at start of each test iteration	Select to clear shared pool between iterations. ℹ **Note:** To perform this action, the Oracle database account must have the ALTER SYSTEM privilege.

ℹ **Note:** Cached data can improve performance, so selecting one or both of these options can prevent cached data from affecting subsequent iterations.

Performance Counters

Select the **Performance Counters** tab of the Connection dialog to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Clustering

Select the **Clustering** tab of the Connection dialog to enable clustering. See Oracle Clustering Tab (Connections) on page 56 for more information.

Statistics

Select the **Statistics** tab of the Connection dialog to specify statistics collection options for this connection. See Oracle Statistics Tab (Connections) on page 56 for more information.

ℹ **Note:** To review or modify a connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Create SQL Server Connection

Benchmark Factory allows you to create a connection to a Microsoft SQL Server database using native database provider connectivity or ODBC connectivity.

To create a SQL Server native connection

ℹ **Note:** For a list of operating systems that support the SQL Server native provider distributed with Benchmark Factory (native client 11.0), see https://msdn.microsoft.com/en-us/library/cc280356(v=sql.120).aspx. If your client operating system is not supported, use ODBC connectivity.

1. Click New Connection in the main toolbar to open the New Connection dialog.
2. Select *Microsoft SQL Server* from the **Database Type** drop-down list.
3. Select the **Native** tab.
4. Enter the following connection information.

Native tab	
Server Name	Enter the name or the IP address of the server. Click the drop-down arrow to retrieve a list of servers running SQL Server that are currently active on the network.
Authentication	Select the type of authentication to use.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection in the My Connections pane.
Reconnect Options	Click **Reconnect Options** to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

ℹ **Note:** BFScripts have been enabled on the User Name and Password fields.

5. Click **Create Connection** to save the connection information and connect.
6. After creating a new SQL Server connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

To create a SQL Server ODBC connection

1. Click New Connection in the main toolbar to open the New Connection dialog.
2. Select *Microsoft SQL Server* from the **Database Type** drop-down list.
3. Select the **ODBC** tab.

4. Enter the following connection information.

ODBC tab	
Data Source Name	The name of the MS SQL Server ODBC data source.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection in the My Connections pane.
Reconnect Options	Click **Reconnect Options** to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

Note: BFScripts have been added to the Data Source name field in the ODBC Connection dialog. BFScripts have been enabled on the User Name and Password fields.

5. Click **Create Connection** to save the connection information and connect.

6. After creating a new SQL Server connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

Miscellaneous Options

Select the **Miscellaneous** tab of the Connection dialog to specify options for flushing cached data. Review the following for additional information.

Note: The Miscellaneous tab is only available when creating a SQL Server Native connection. For a SQL Server ODBC connection, the default options specified in **Edit | Settings | SQL Server** are used.

Database Flush	
Clean data buffer and procedure caches at start of each test iteration	Select this option to instruct Benchmark Factory to clear cached data between iterations. Cached data can improve performance, so selecting this option can prevent cached data from affecting subsequent iterations. **Notes:** • This option is only applicable to SQL Server 2005 or later. • To perform this action, the SQL Server database account must have the sysadmin fixed server role.

Performance Counters

Select the **Performance Counters** tab of the Connection dialog to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Note: To review or modify an existing connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Create SAP Connection

Benchmark Factory allows you to create a native SAP database connection.

To create an SAP database connection

1. Click New Connection in the main toolbar. The New Connection dialog opens.
2. Select *SAP* from the drop-down list.
3. Enter the connection information. Review the following for additional information:

   Note: BFScripts have been enabled on the User Name and Password fields.

Native	
Server Name	The name or the IP address of the server.
Database	The name of the database to which you wan to connect.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection in the My Connections pane.
Reconnect Options	Click link to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

4. Click **Create Connection** to save the connection information and connect.
5. After creating a new SAP connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

Performance Counters

Benchmark Factory allows you to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Note: To review or modify a connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Create MySQL Connection

Benchmark Factory allows you to create an ODBC connection to a MySQL database.

To create a MySQL connection

1. Click New Connection in the main toolbar. The New Connection dialog opens.
2. Select *MySQL (ODBC)* from the drop-down list.
3. Enter the connection information. Review the following for additional information:

   Note: BFScripts have been added to the Data Source name field in the ODBC Connection dialog. BFScripts have been enabled on the User Name and Password fields.

ODBC	
Data Source Name	The name of the MySQL ODBC data source.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection in the My Connections pane.
Reconnect Options	Click link to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

4. Click **Create Connection** to save the connection information and connect.
5. After creating a new MySQL connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

Performance Counters

Benchmark Factory allows you to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Note: To review or modify a connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Create ODBC Connection

Benchmark Factory supports almost all databases that you can connect to using an ODBC 3.0 or later driver. Create an ODBC connection to access these databases.

To create an ODBC connection

1. Click New Connection in the main toolbar. The New Connection dialog opens.
2. Select *ODBC* from the drop-down list.
3. Enter the connection information. Review the following for additional information:

   Note: BFScripts have been added to the Data Source name field in the ODBC Connection dialog. BFScripts have been enabled on the User Name and Password fields.

ODBC	
Data Source Name	The name of the ODBC data source.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection in the My Connections pane.
Reconnect Options	Click link to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

4. Click **Create Connection** to save the connection information and connect.

Performance Counters

Benchmark Factory allows you to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Note: To review or modify a connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Create IBM DB2 CLI Connection

Benchmark Factory allows you to create a native connection to an IBM DB2 database.

To create an IBM DB2 CLI connection

1. Click New Connection in the main toolbar. The New Connection dialog opens.
2. Select *IBM DB2 CLI* from the drop-down list.
3. Enter the connection information. Review the following for additional information:

   Note: BFScripts have been enabled on the User Name and Password fields.

Native	
Database Alias	Enter the database name.
User Name	Enter the user name to use for this connection.
Password	Enter the password associated with the user name.
Connection Name	Enter a name to use to identify this connection.
Reconnect Options	Click link to specify timeout and reconnect options. See Connection Timeout and Reconnect Options.

4. Click **Create Connection** to save the connection information and connect.
5. After creating a new DB2 connection, you can collect database and host server information using the Edit Connection dialog. See Environment Information on page 54 for more information.

Performance Counters

Benchmark Factory allows you to add additional performance counters to a connection. See Performance Counters Tab on page 111 for more information.

Note: To review or modify a connection, click **Edit Connections** in the main toolbar. Select a connection and click .

Edit Connections

To edit a connection, use the **My Connections** pane or click the **Edit Connections** button in the main toolbar. Both the My Connections pane and the Edit Connections dialog provide a list of your currently-defined connections. Use either of these interfaces to view or modify information for each connection.

To edit a connection

1. Use one of the following methods to edit a connection:
   - In the **My Connections** pane, select a connection and click Edit.
   - Click Edit Connections in the main toolbar. Then select a connection and click in the Edit Connections dialog.
2. The Connection dialog for the selected connection opens. Use the DB Connection tab to update the connection password or other connection information.
3. Select from among the other tabs to add or modify other properties associated with the connection. The following tabs are available (depending on the connection type).
   - Environment Information
   - Performance Counters Tab
   - Oracle Clustering Tab (Connections)
   - Oracle Statistics Tab (Connections)

Connection Toolbars

Edit Connections Dialog Toolbar

The following commands are available from the toolbar in the **Edit Connections** dialog.

	Open the New Connection dialog to create a new connection.
✕	Remove the selected connection.
	Open the Connection dialog for the selected connection. Use this dialog to view or edit connection properties.

My Connections Pane Toolbar

The following commands are available from the toolbar in the **My Connections** pane.

New	Create a new connection.
Edit	Edit the selected connection.
Duplicate	Duplicate the selected connection.
✕ Delete	Delete the selected connection.

Connection Timeout and Reconnect Options

You can specify timeout and reconnect options when you are creating a new connection. You can also specify these options at any time for an existing connection.

To specify timeout and reconnect options

1. In the New Connection dialog, after entering connection information, click **Reconnect Options**.
2. Specify options. Review the following for additional information:

Timeout	
Time	The maximum amount of time Benchmark Factory will try to log on to the system-under-test. If this amount of time is reached, Benchmark Factory will return an error.
Infinite timeout	Prevents the logon to the system-under-test from timing out and returning an error.
Reconnect	
Enable Reconnect	Enables Benchmark Factory to attempt to reconnect to the system-under-test if the connection is lost.
Number of reconnect attempts	The number of times to attempt to reconnect before aborting.
Time to wait between reconnect attempts (seconds)	How long to wait before attempting to reconnect.

3. Click **OK**.

To specify timeout and reconnect options for an existing connection

1. Click Edit Connections in the main toolbar.
2. Select the connection and click .
3. Then click **Reconnect Options** in the Connection dialog.

Environment Information

Benchmark Factory can collect and display database and host server information for each connection. The information is displayed in the **Environment** tab of the Connection dialog for an existing connection and also in the **Database Under Test** page of the New/Edit Job Wizard.

You can also create custom properties to add your own customized information to the connection.

To display environment information for an existing connection

1. Navigate to the Connection dialog for an existing connection using one of the following methods:
   a. In the **My Connections** pane, select a connection and click Edit.
   b. Click Edit Connections in the main toolbar. Select a connection and click in the Edit Connections dialog.
2. Select the **Environment** tab in the Connection dialog.
3. Click **Detect Environment Information**.

   Note: To successfully view all environment information requested by Benchmark Factory, the login account used in the connection must have sufficient permissions.

To display environment information in the New/Edit Job wizard

1. Open the New Job Wizard or select a job and open the Edit Job wizard.
2. Select the **Database Under Test** page.
3. (New job) Select a connection.
4. Click **Detect Environment Information**.

   Note: To successfully view all environment information requested by Benchmark Factory, the login account used in the connection must have sufficient permissions.

To create a custom property

1. In the Connection dialog for a connection, select the **Environment** tab.
2. Click **Add**.
3. Enter a property name and value. Click **OK**.
4. The new property and value display in the Environment tab under **Custom Properties**.

Permissions Required to Detect Environment Information

In order to display database and host server information for a particular connection, the user account for that connection must have the necessary permissions. Review the following special requirements for detecting environment information.

Connection Type	Required Permissions
SQL Server	To view Host CPU Count and Host Memory, the login account requires the **VIEW SERVER STATE** permission

Performance Counters Tab

Performance counters keep track of statistics during test execution. By default the following performance counters are recorded when a job executes, even though they do not display in the Performance Counter tab for a connection or job.

- AVG_TIME
- BPS
- DEADLOCKS
- TOTAL_ERRORS
- MAX_TIME
- RPS
- TOTAL_BYTES
- TOTAL_ROWS
- TPS
- USERLOAD

You can add performance counters to a connection or a job.

To add performance counters to a connection

1. In the New Connection dialog when creating a new connection (or the Connection dialog for an existing connection), select the Performance Counters tab.
2. To add a new counter, click .
3. Select a computer from the drop-down list.
4. To display counters, expand an object node by clicking the node's down-arrow.
5. Select one or more counters and click **Add**.
6. When you have finished adding counters, click **Close** to close the Browse Performance Counters dialog.
7. To delete a counter, select the counter and click ✖.

Oracle Statistics Tab (Connections)

Benchmark Factory allows you to specify statistics collection options for each Oracle connection.

To specify statistics collection options for an Oracle connection

1. In the New Connection dialog when creating a new connection (or in the Connection dialog for an existing connection), select the **Statistics** tab.
2. Select one of the following:
   - **Statspack Options**—Collect database-wide performance statistics using the Oracle Stats Pack utility.
   - **AWR and ADDM Options**—Collect database-wide performance statistics using the optional Oracle Enterprise Manager (OEM) “Diagnostic Pack” Automatic Workload Repository (AWR).

     Note: A valid licenses is required to use the optional OEM Diagnostics Pack.

Tip: You can specify default values for Oracle performance collection and reporting options in **Edit | Settings | Oracle**.

Oracle Clustering Tab (Connections)

Clustering is the process of using two or more computer systems that work together. Multiple servers are linked to handle variable workloads or to provide continued operation in the event one fails. Computers may be multiprocessor systems. A cluster of four computers with four CPUs each provides a total of sixteen CPUs processing at one time.

Use the Clustering tab in the Connection dialog of an Oracle connection to enable clustering. This allows you to perform Oracle Real Application (RAC) Testing.

To enable clustering for an Oracle connection

1. In the Connection dialog, after entering connection information, select the Clustering tab.
2. Select the **Enable Clustering** checkbox.
3. Specify the number of nodes and then the percentage of users that will hit a particular node. See Real Application Clustering (RAC) Load Testing on page 57 for more information.
4. Click **OK** to save your changes.

Real Application Clustering (RAC) Load Testing

Oracle uses clustering to allow one database to span multiple resources. Benchmark Factory can simulate user activity on these systems.

Performing Oracle RAC testing in Benchmark Factory requires the following steps:

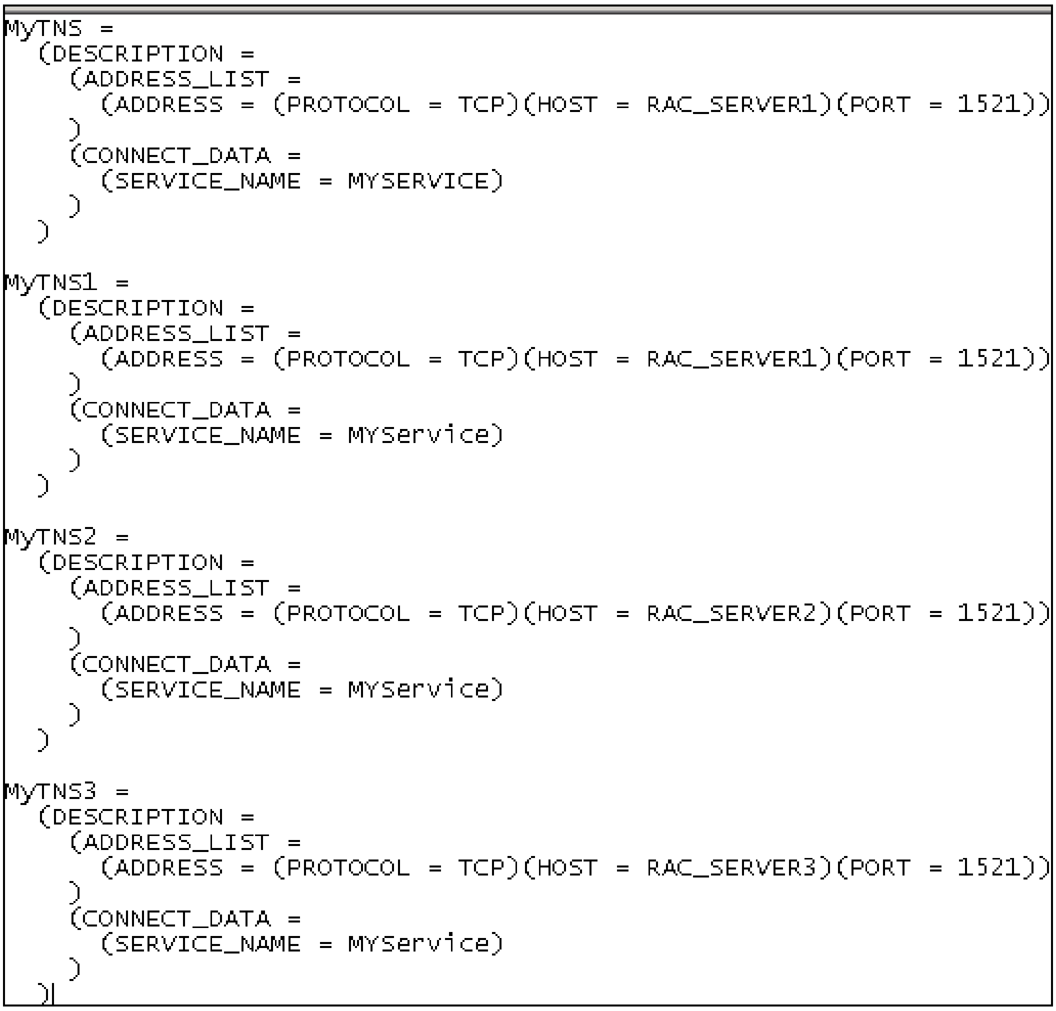
- Setting up the tnsnames.ora file
- Creating the Benchmark Factory cluster profile

Setting up the tnsnames.ora file

1. Navigate to the **tnsnames.ora** file located in the Oracle directory. Transparent Network Substrate (TNS) is Oracle's networking architecture. TNS provides a standard application interface that enables network applications to transparently access underlying network protocols.
2. Open the file in a text editor. Edit the **tnsnames.ora** file to set up clustering connections.
    a. For the first entry, the Net Service Name (in this example MyTNS) is used to define the root Net Service Name. The root Net Service Name is not used when running a test.

    b. The remaining Net Service Names must be named with the root and a number. In this example, MyTNS is the root and MyTNS1, MyTNS2, and MyTNS3 will be used when the test is run. The "MyTNS" and "MyTNS1" connections are duplicates and are required for Benchmark Factory to perform load testing. “MyTNS2” and “MyTNS3” are for the other two machines in the cluster.

    The following provides an example of a **tnsnames.ora** file set up for a three node cluster.



```
MyTNS =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = RAC_SERVER1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = MYSERVICE)
    )
  )

MyTNS1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = RAC_SERVER1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = MYService)
    )
  )

MyTNS2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = RAC_SERVER2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = MYService)
    )
  )

MyTNS3 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = RAC_SERVER3)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = MYService)
    )
  )
```

Creating the Benchmark Factory clustering connection

1. Click New Connection in the main toolbar.
2. Select *Oracle* from the drop-down list.
3. Enter the connection information. See Create Oracle Connection on page 46 for more information.

   Note: The Net Service Name reflects the same entry as the first entry in the tnsnames.ora file.

4. After entering the connection information, select the Clustering tab.
5. Select the **Enable Clustering** checkbox. The number of default nodes displayed is 2.
6. Enter the desired number of nodes in the **Number of Nodes**. For this example, we have 3 nodes. The following provides the mapping structure from the Benchmark Factory nodes to the **tnsnames.ora** file.
   - Node 1 = MyTNS1
   - Node 2 = MyTNS2
   - Node 3 = MyTNS3
7. Next, you need to set up the percentage of users that will hit a particular node. Click inside the Percentage Column for each node and enter the desired percentage of users that will hit that particular node.
8. Click OK to save your changes. The Oracle RAC Clustering has been set up and is ready to run.

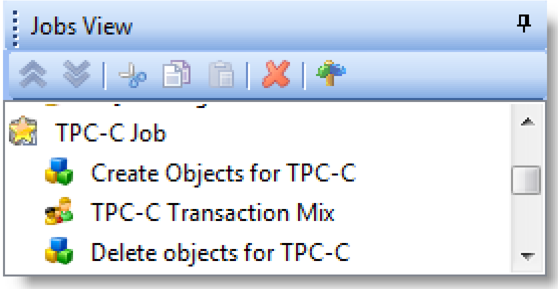
8

Create and Edit Tests and Jobs

Jobs View

The Jobs View displays a list of saved jobs, shows which jobs are currently running, and which jobs are scheduled to run.

From the Jobs View pane, you can run jobs, view job status, and view real-time testing results.



To run a job

- Select the job in the Jobs View pane and click Run Job.

To view test results

- To view a job's test results, select a job in the Jobs View pane. Test results display in the right pane. See Benchmark Factory Console for an overview of the Benchmark Factory console.
- To compare two or more run results for a test, select the **Compare Results** tab. Use **Ctrl+click** to select multiple test runs. A comparison of the results for the various runs displays.

To edit an existing job

- To edit an existing job (to modify the workload or job setup options), select the job in the Jobs View pane and click Edit Job.

View Job Status

From the Jobs View pane, you can view job status.

The following job states are identified:

- **Scheduled**: All jobs currently waiting to run or scheduled to run at a future time.
- **Running**: Job currently running.
- **Completed**: All completed jobs.

Jobs View Toolbar

The Jobs View toolbar provides user functionality to save or delete jobs.

⏫	Move the selected test up in the list.
⏬	Move the selected test down in the list.
✖	Delete the selected item from the Jobs View.
🌳	Toggles tree-view on and off.

Note: To save a job as a Benchmark Factory script, select the job and click **Save** in the Benchmark Factory toolbar or select **File | Save**.

Quickstart: Create a New Job

Use the New Job Wizard to create a new job. The New Job Wizard guides you through the process of creating a database load test (workload), as well as selecting a database to test, setting general job options, and specifying agents.

Navigating the New Job Wizard is easy. The three major steps are displayed in the left pane: Database Under Test, Workload, and Agent. The individual tests you add are displayed under the Workload step. Click on a step or test in the left pane to display the available options and properties in the right pane.

To allow the wizard to guide you through each step, after completing a page click the green arrow at the bottom of the page. This will advance you to the next step. For example, after selecting a connection on the first page of the wizard, click Add Workload.

To create a new job

1. Click New in the Benchmark Factory main toolbar. The New Job Wizard opens.
2. **Select a connection.** Select a connection from the drop-down list in the **Database Under Test** page.
    - Click **New** to define a new connection. See About Benchmark Factory Connections.
    - Click **Edit** to edit the selected connection. See Edit Connections.
    - Click **Test** to test the selected connection.
3. The Database Under Test page also provides database and host information for the selected connection.
    - To refresh this information, click **Detect Environment Information**.
    - You can edit this information or add custom properties. This information is also displayed in the **Environment** tab of the Connection dialog. See Environment Information.
4. **Add Workload.** After selecting a database connection, click Add Workload at the bottom of the Database Under Test page, or click **Workload** in the left pane,.
5. **Select a test.** On the workload page, select the type of test to perform from the drop-down list. Then select a test to add to the workload. To learn how to create a specific test, select from the following:

- ***Industry Standard Benchmark Test***—These tests simulate real-world application workloads. Select from a number of standard benchmarks included with Benchmark Factory.
    - For benchmark test descriptions, see Overview of Benchmark Testing.
    - To create a benchmark test, see Create Industry Standard Benchmark Test.
- ***Capture / Replay Test***—Capture and replay a workload; replay a workload from a trace file; or import a workload for a database repository, text file, or export file from another Quest Software tool.
    - To create a capture/replay test, see Capture and Replay Tests.
- ***Artificial Test for Desired Effect***—Design a workload using a mix of benchmark tests, transaction types, SQL statement types, or resource loads.
    - To create an artificial test, see Artificial Test for Desired Effect.
- ***Scalability Test***—Use this type of test to load test and compare the performance of different SQL statement variations.
    - To create a scalability test for SQL, see Create SQL Scalability Test.
- ***Custom Test***—To create a custom Mix test, Replay test, Goal test, or SQL Scalability test, see Custom Tests.
- ***Create/Delete Benchmark Objects Test***—To add a step to create or delete benchmark objects, see Add a Create/Delete Benchmark Objects Test.
- ***Execute External File***—To add a step to execute a file, see Execute External File.

6. Click **Add Test** or **Select Test** at the bottom of the page to add the test to the workload.
7. **Specify test options.** Specify test options for the selected test. Select one of the links above for detailed information about test options for each test type.
8. **Job Setup.** (Optional) You can specify job-level options. Click **Job Setup** at the bottom of the Test Options tab, or select the **Job Setup** tab. See Job Setup Tab to learn more.
9. **Agent Setup.** After specifying options for the test you selected, set up the agents for this job.
    a. Click the **Agent Setup** button at the bottom of the page, or click **Agent** in the left pane.
    b. Click inside the checkbox to the left of the agent name to select it. A checkmark displays for each selected agent. See About Agents on page 36 for more information.

       Note: When you run a job using one or more local agents, if Agent utilization of resources on the local machine is too high, errors could occur.
10. After specifying the test-level and job-level options, you can save the job, run the job, or schedule the job. Review the following:
    - To run the job immediately, click Run Job.
    - To schedule the job, click **Workload** (left pane) and select the **Job Setup** tab. Then select the **Schedule** tab. See Schedule a Job.
    - To save the job without running it immediately, click Save / Close. Use this option after scheduling a job.
    - To save the job to an existing job, select the **Job Setup** tab. In the Save Job section, select the name of an existing job. Click Save / Close.

Notes:

- After creating a job, you can save it as a Benchmark Factory script. Select the job in the Jobs View pane and click **Save** in the main toolbar or select **File | Save**.
- To modify an existing job, right-click the job and select **Edit Job**.
- After creating a job, you can add tests to the workload. Right-click the job and select **Edit Job**.

Edit a Job

Use the Edit Job Wizard to modify an existing job.

To edit a job

1. Select a job in the Jobs View and click Edit Job.
2. See the topic on the test type you want to edit for more information on editing test and job setup options. See Quickstart: Create a New Job on page 60 for more information.

Industry Standard Benchmark Tests

Create Industry Standard Benchmark Test

This test type allows you to create load scenarios from one of the following industry standard benchmark tests: TPC-C, TPC-E, TPC-H, AS3AP, Scalable Hardware, TPC-B, or TPC-D. Each of these tests includes the steps to create/load all the required benchmark objects needed to execute the standard benchmark workloads. These synthetic workloads can be used when no real-world data is available to import for database load testing.

This test type also allows you to create a workload using the Benchmark Factory Replication test.

For more information, including a description of each of the standard benchmark tests included in Benchmark Factory, see Overview of Benchmark Testing.

To create an Industry Standard Benchmark test

1. Select **File | New** to open the New Job Wizard.
2. Select a connection. Then click **Workload** in the left pane.
3. In the Workload page, select **Industry Standard Benchmark Test** from the test type drop-down list.
4. Then select the type of benchmark test you want to use. Select each test to display a description in the lower pane.
5. **Number of streams to include in the Stream Test.** (TPC-H only) Select to include a Stream Test in a TPC-H test. Then specify the number of streams to include.
6. Click Add Test to add the test to the workload. The Summary tab opens.
7. **Summary tab.** The Summary tab provides a summary of the job and the workload, as well as links to the commonly edited options for this test. Click each link to navigate to the applicable tab where you can edit that option. Options shown in red are required. Review the following for additional information:

Scale	Click to change the scale factor for this test/step. In the **Benchmark Scale** field, specify a scale factor. See Benchmark Scale Factor on page 134 for more information.
Size	Displays the total size of all objects in this Create Objects step. Click to open the Scale tab where you can modify the database size or the scale factor. See Benchmark Scale Factor on page 134 for more information.
Number of Tables	(Replication test only) Displays the number of tables to create. Click to modify the number of tables, the number of columns in a table, and the

	data types to create. See Replication Table Options Tab on page 97 for more information.
Transactions	Displays the number of transactions. Click to modify the transaction mix for the transaction step. See Transactions Tab on page 98 for more information.
User Load	Displays the user load—the number of virtual users per test iteration. Click to review or modify the user load. See Specify User Load on page 134 for more information.
Length	Click to modify the timing for this test. See Timing Tab on page 100 for more information.

8. **Create Objects - More Options.** To specify more options for the Create Objects step, select the **Create Object for *test*** step in the left pane. Then select the **Test Options** tab. Review the following for additional information:

Scale tab	See Benchmark Scale Factor.
Options tab	Use the Options tab to specify some general options for this step, such as specifying the number of agents to use to load data. See Options Tab (Create Objects Step).
Custom Creation SQL tab	Use the Custom Creation SQL tab to customize the SQL used to create objects in this step. See Customize SQL for Creating Objects.

9. **Transaction Mix - More Options.** To specify more options for the Transaction Mix step, select the ***test* Transaction Mix** step in the left pane. Then select the **Test Options** tab. Review the following for additional information:

Transactions tab	See Transactions Tab.
User Load tab	See Specify User Load.
Timing tab	See Timing Tab.
Advanced tab	See Advanced Tab.

10. To add another test to the workload, click **Add Another Test/Step**. This takes you back to the selection page of the **Workload** section.

11. **Job Setup.** To configure job-level options, select the **Job Setup** tab. See Job Setup Tab to learn more about job-level options.

12. **Agent.** To set up the agents for this job, click **Agent** in the left pane. Select the agents/computers to use for this test. See About Agents on page 36 for more information.

13. After specifying the test-level and job-level options, you can save the job, run the job, or schedule the job. Review the following:

    - To run the job immediately, click Run Job.
    - To schedule the job, click **Workload** (left pane) and select the **Job Setup** tab. Then select the **Schedule** tab. See Schedule a Job.
    - To save the job without running it immediately, click Save / Close. Use this option after scheduling a job.
    - To save the job to an existing job, select the **Job Setup** tab. In the Save Job section, select the name of an existing job. Click Save / Close.

Create a Replication Benchmark Test

The Replication benchmark test is a unique test developed by Benchmark Factory for testing replication in applications such as SharePlex.

This test allows you to create customized database objects, as well as a customized workload, in order to produce a repeatable database change rate and history (for example, Oracle redo logs) that can be used to test replication strategies. This test can also be used to create a model of your database and workload in order to help demonstrate replication performance.

While this benchmark was designed for testing and evaluating replication strategies and functionality in SharePlex, it can also be used to test other database replication products as well.

The following options can be customized in the Replication benchmark test:

- The number of tables in the test
- The number of columns per table and the column types (data types)
- The percentage of each column type (data type) in all tables, for example, 40% INT, 20% VARCHAR (255), etc.
- The percentage of each statement type (insert, update, delete), as well as the number of statements per commit (transaction)

To create a Replication benchmark test

1. Select **File | New** to open the New Job Wizard.
2. In the New Job Wizard, select a connection in the **Database Under Test** page. Then click **Workload** in the left pane.
3. In the Workload page, select **Industry Standard Benchmark Test** from the test type drop-down list.
4. Then select the **Replication** test from the Other Tests category.
5. Click Add Test to add the test to the workload. The Summary tab opens.
6. **Summary tab.** The Summary tab provides a summary of the job and the workload, as well as links to the commonly edited options for this test. Click each link to navigate to the applicable tab where you can edit that option. Options shown in red are required. Review the following for additional information:

Scale	Click to change the scale factor for this test/step. In the **Benchmark Scale** field, specify a scale factor. See Benchmark Scale Factor on page 134 for more information.
Size	Displays the total size of all objects in this Create Objects step. Click to open the Scale tab where you can modify the database size or the scale factor. See Benchmark Scale Factor on page 134 for more information.
Number of Tables	(Replication test only) Displays the number of tables to create. Click to modify the number of tables, the number of columns in a table, and the data types to create. See Replication Table Options Tab on page 97 for more information.
Transactions	Displays the number of transactions. Click to modify the transaction mix for the transaction step. See Transactions Tab on page 98 for more information.
User Load	Displays the user load—the number of virtual users per test iteration. Click to review or modify the user load. See Specify User Load on page 134 for more information.
Length	Click to modify the timing for this test. See Timing Tab on page 100 for more information.

7. **Create Objects - More Options.** To specify more options for the Create Objects step, select the **Create Object for Replication** step in the left pane. Then select the **Test Options** tab. Review the following for more information:

Scale tab	See Benchmark Scale Factor.
Table Options tab	Use the Table Options tab to specify options such as the number of tables, number of columns, column data types, and data type percentages. See Replication Table Options Tab.
Options tab	Use the Options tab to specify some general options for this step, such as specifying the number of agents to use to load data. See Options Tab (Create Objects Step). **Object/Data Retention Method - Special Considerations** • The **Recreate objects and reload data every execution** method will override the selected method in the following scenarios: • If you modify the list or proportion of data types or the number of columns between runs • If you select **Refresh data using inserts** and modify the number of tables between runs • Tables are added or deleted if you select either **Refresh data using backup sets** or **Create/Load objects if objects don't exist** and modify the number of tables between runs. • Rows are added or deleted if you select **Refresh data using backup sets** and modify the scale factor between runs.

8. **Replication Test - More Options.** To specify more options for the Replication step, select the **Replication Test** step in the left pane. Then select the **Test Options** tab. Review the following for more information:

Transactions tab	See Transactions Tab.
User Load tab	See Specify User Load.
Timing tab	See Timing Tab.
Advanced tab	See Advanced Tab.

9. To add another test to the workload, click **Add Another Test/Step**. This takes you back to the selection page of the **Workload** section.

10. **Job Setup.** To configure job-level options, select the **Job Setup** tab. See Job Setup Tab to learn more about job-level options.

11. **Agent.** To set up the agents for this job, click **Agent** in the left pane. Select the agents/computers to use for this test. See About Agents on page 36 for more information.

12. After specifying the test-level and job-level options, you can save the job, run the job, or schedule the job. Review the following:

    - To run the job immediately, click Run Job.
    - To schedule the job, click **Workload** (left pane) and select the **Job Setup** tab. Then select the **Schedule** tab. See Schedule a Job.
    - To save the job without running it immediately, click Save / Close. Use this option after scheduling a job.
    - To save the job to an existing job, select the **Job Setup** tab. In the Save Job section, select the name of an existing job. Click Save / Close.

Notes:

- Each Insert statement is applied to only one row.
- Each Update or Delete statement is applied to only one row if the target row exists. If the target row does not exist, no rows are affected.
- In a Replication test, each incremental increase in the scale factor loads an additional 1000 rows of data into each table.

Capture and Replay a Workload

Capture and Replay Tests

Capture/Replay tests allow you to capture and then replay a workload, or replay an existing capture from a trace file, database repository, text file, or output from another Quest Software tool.

Select one of the following topics for more information:

- Capture and Replay Oracle Workload
- Capture and Replay SQL Server Workload
- Replay Load from an Oracle Trace File
- Replay from SQL Server Trace Table
- Replay Load from an ODBC Trace File
- Import from a Delimited Text File
- Import from Oracle Dynamic Performance View
- Import from Quest Tools Export File

Capture and Replay Oracle Workload

Benchmark Factory allows you to capture Oracle workloads, then replay those workloads to view transactions and response times.

In Benchmark Factory, use one of the following workflows to capture an Oracle workload:

- **New Job Wizard**—Use the New Job Wizard (**File | New**) to guide you through the steps of setting up and capturing the workload. Then add the captured workload to your new job to run it immediately or at a later time.
- **Capture Scenario Wizard**—Select **Tools | Capture Workload** to open the Capture Scenario Wizard.

Note: This feature is not available in the freeware edition of Benchmark Factory.

See also, Considerations When Performing an Oracle Capture and Replay.

To capture an Oracle workload

1. Select **File | New** to open the New Job Wizard.
2. On the **Database Under Test** page, select the database to test.
3. Then click Add Workload.
4. On the Workload page, select **Capture/Replay Test** from the drop-down list.
5. Then select the **Capture and Replay Oracle Workload** option.

6. Click Select Test.

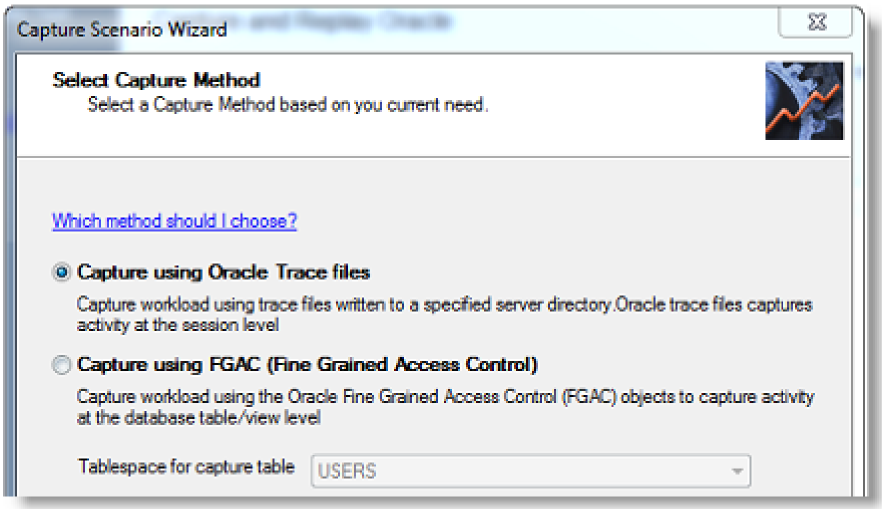
7. On the Replay Workload page, click **Perform New Capture**. The Capture Scenario Wizard opens and the New Job Wizard closes.

   Tip: You can also create a new capture using the same settings as an existing capture project. In the Captures tab of the Benchmark Factory console, right-click a capture and select **Repeat Capture**.

8. **Capture Scenario Wizard.** In the Capture Scenario Wizard, enter connection information for the database from which you want to capture a workload. For more information, see Create Oracle Connection. Click **Next** when finished.

9. **Select Capture Method.** On the Select Capture Method page, select a capture method to use. Review the options below. Click **Next** when finished.

Capture using Oracle Trace files	Captures workloads using trace files written to a specified server directory. These files are captured at the session level. **Note:** The Trace file capture method is not available for Oracle connections that use Real Application Clusters (RAC).
Capture using FGAC (Fine-Grained Access Control)	Captures workloads using the Oracle Fine-Grained Access objects that capture activities at the database table and view levels. **Tablespace for capture table**—Allows you to select the required tablespace for Fine Grained Access captures. **Note:** The FGAC capture method is enabled only if the feature is available in the target database.



Capture Scenario Wizard

Select Capture Method

Select a Capture Method based on you current need.

Which method should I choose?

● **Capture using Oracle Trace files**

Capture workload using trace files written to a specified server directory.Oracle trace files captures activity at the session level

○ **Capture using FGAC (Fine Grained Access Control)**

Capture workload using the Oracle Fine Grained Access Control (FGAC) objects to capture activity at the database table/view level

Tablespace for capture table: USERS

10. **Apply Privileges.** If the user does not have the required privileges to do the capture, the Apply Privileges page opens. Enter the credentials of a DBA-type user account that can apply the necessary privileges. Click **Next** when finished.

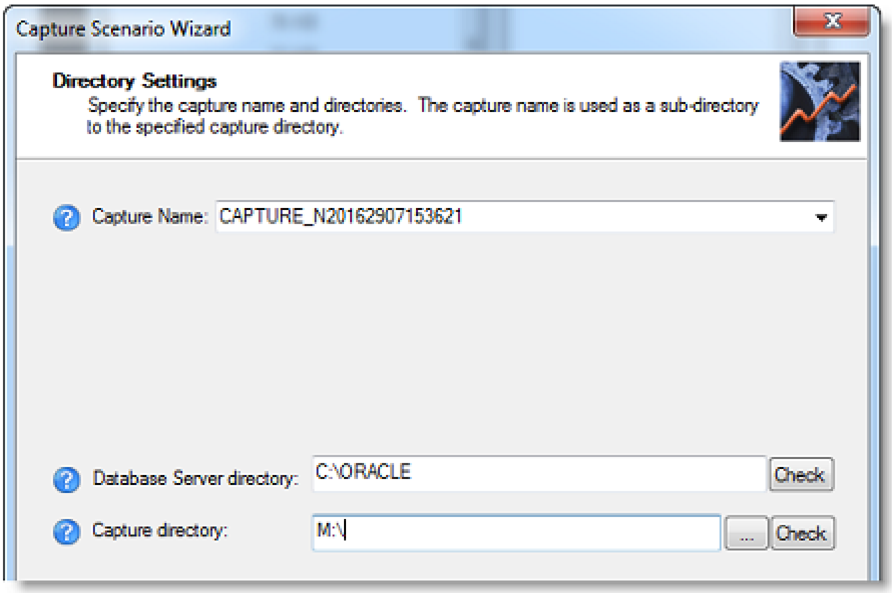
    Note: To view the missing privileges necessary for this user to perform the capture process, click the **View/Save Script** button and review the script.

11. **Directory Settings.** On the Directory Settings page, specify an Oracle server-side directory in which to place the capture files. Also, specify the capture directory from which Benchmark Factory will replay the files. Review the following for additional information:

    Click **Next** when finished.

Capture Name	Enter a name for the capture, or use the default. This name is used for the

	sub-directory where the capture files are stored in the Capture Directory you specify. ℹ **Note:** Only alpha and numeric characters and the underscore (_) are permitted. The name must begin with a letter.
Database Server Directory	Specify an Oracle server-side directory where Benchmark Factory should place the capture files. Specify the path as the server sees it. ℹ **Note:** You can specify a network directory here. Enter the full network path to the network directory. The database service must be able to access the network directory. In a Linux environment, a local directory must be mounted to the network location.
Capture Directory	Specify a directory where Benchmark Factory will look for the capture files to replay. Do one of the following: • **Manual transfer:** Specify a client-side directory and then manually transfer the capture files to this location. • **Shared directory:** Specify the same Oracle server-side (or network) directory you specified in the preceding field. However, enter the path as the client computer sees it. To use this method, you must first map a drive on the client computer to the Database Server Directory (Oracle server-side or network directory). Then, browse to and select that mapped drive. **Selecting a shared directory:** The shared capture directory can be on the database server as a local directory or on a network file server as a local directory. Either location must be accessible by the Benchmark Factory client computer (Windows network share, Samba, or NFS). A network file server is the preferred location for the following reasons: • This minimizes total network I/O required for capture and replay. • The shared directory can be placed on an I/O subsystem with sufficient I/O bandwidth to handle both the concurrent I/O writing and the cumulative size of the trace and export files.



Directory Settings

Specify the capture name and directories. The capture name is used as a sub-directory to the specified capture directory.

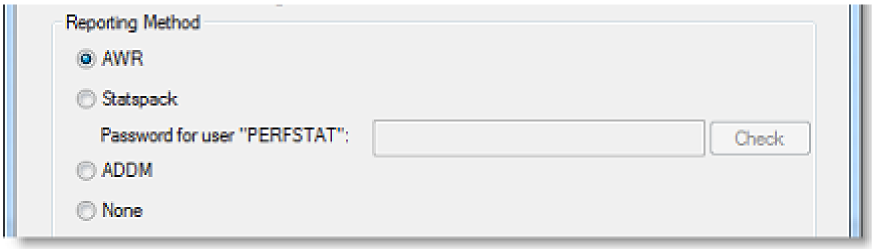
Capture Name: CAPTURE_N20162907153621

Database Server directory: C:\ORACLE

Capture directory: M:\

12. **Reporting method.** On the Reporting Settings page, select the performance reporting method to use. Review the options below. Click **Next** when finished.

Reporting Method	Select a performance collection and reporting method. • **AWR**—If this option is selected, Benchmark Factory uses the optional Oracle Enterprise Manager (OEM) “Diagnostic Pack” Automatic Workload Repository (AWR) to collect database wide performance statistics, collect before and after workload capture, replay AWR snapshots of database-wide performance statistics, and generate a differences report between the capture and replay results. ℹ **Note**: A valid licenses is required to use the optional OEM Diagnostics Pack. • **Statspack**—If this option is selected, Benchmark Factory calls the Oracle “Stats Pack” utility to create a performance data repository, collect before and after workload capture, replay Stats Pack snapshots of database-wide performance statistics, and generate a different report between the capture and replay results. ℹ Note: "perfstat" is the default password. • **ADDM**—Select to instruct Benchmark Factory to use performance analysis information from the Oracle Automatic Database Diagnostic Monitor (ADDM). • **None**—If this option is selected, no reports are generated.



Reporting Method

● AWR

○ Statspack

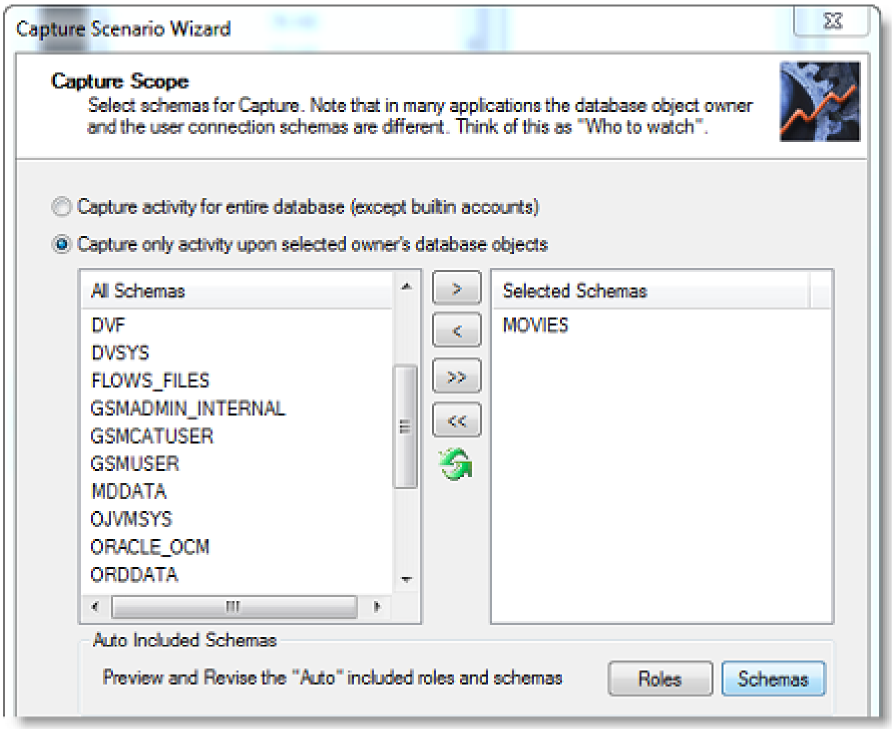
Password for user "PERFSTAT":

○ ADDM

○ None

13. **Capture Scope.** The Capture Scope page allows you to select schemas for capture. You can select the entire database or use the list to select specific schemas. Review the options below. Click **Next** when finished.

Capture activity for entire database	Captures activity for the entire database, but not for built-in accounts.
Capture only activity upon selected owner's database objects	Captures only the data of the selected database objects. This option allows you to select individual schemas for capture. a. To select database objects, move the desired schemas to the right pane. b. To specify auto-included schemas, click one or both of the following buttons: • **Roles**—Opens the Auto Included Roles dialog which displays the roles that can access the objects in the schemas in the Selected Schemas column. Select which roles to include or exclude. • **Schemas**—Opens the Auto Included Schemas dialog which displays schemas that can access the objects in the Selected Schemas. Select which related schemas to include or exclude.



Capture Scenario Wizard

Capture Scope

Select schemas for Capture. Note that in many applications the database object owner and the user connection schemas are different. Think of this as "Who to watch".

○ Capture activity for entire database (except builtin accounts)

● Capture only activity upon selected owner's database objects

All Schemas	Selected Schemas
DVF	MOVIES
DVSYS	
FLOWS_FILES	
GSMADMIN_INTERNAL	
GSMCATUSER	
GSMUSER	
MDDATA	
OJVMSYS	
ORACLE_OCM	
ORDDATA	

Auto Included Schemas

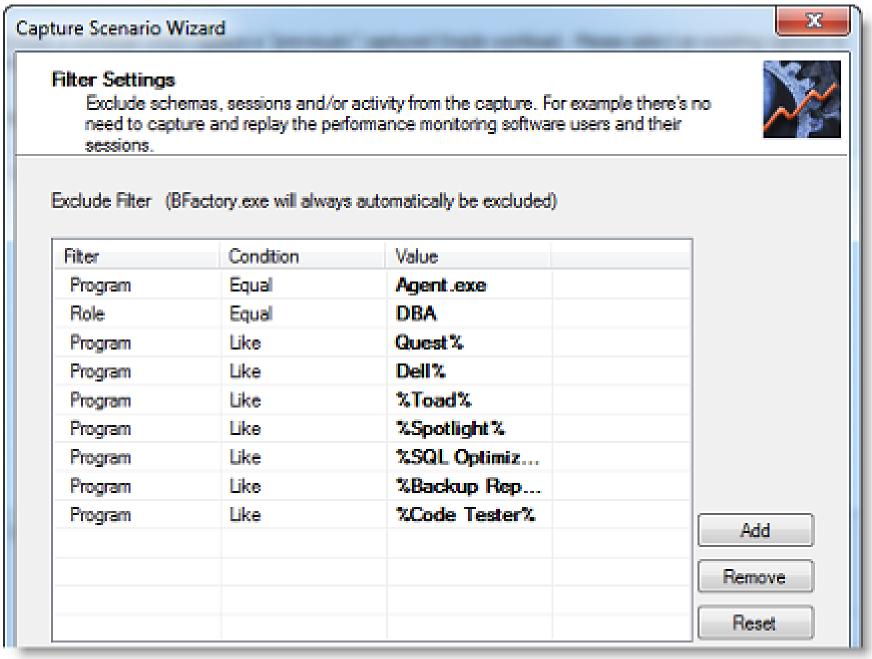
Preview and Revise the "Auto" included roles and schemas

14. **Export Scope.** The Export Scope page allows you to specify whether or not to export the database objects and data required to replay the workload on the same data snapshot. Review the options below. Click **Next** when finished.

Perform export as part of capture process	Select to instruct Benchmark Factory to export the objects selected on the previous page (Capture Scope). The export is performed during the capture procedure. **Note:** If you selected to capture activity for the entire database, exporting the entire database can require significant time and space. **Include export of related schemas**—Select to export the auto-included schemas selected on the Capture Scope page.

15. **Filter Settings.** (Optional) The Filtered Settings page allows you to add filters to exclude activity from the capture.

- To add a filter, click **Add**. Then specify parameters for the filter. Click **Next** when finished.



Capture Scenario Wizard

Filter Settings

Exclude schemas, sessions and/or activity from the capture. For example there's no need to capture and replay the performance monitoring software users and their sessions.

Exclude Filter (BFactory.exe will always automatically be excluded)

Filter	Condition	Value
Program	Equal	Agent.exe
Role	Equal	DBA
Program	Like	Quest%
Program	Like	Dell%
Program	Like	%Toad%
Program	Like	%Spotlight%
Program	Like	%SQL Optimiz...
Program	Like	%Backup Rep...
Program	Like	%Code Tester%

16. **Capture Thresholds.** Benchmark Factory allows you to specify limits for CPU usage and free space during a capture. If levels exceed the values you specify, the capture process is stopped.

    Benchmark Factory displays the current values to help you determine the best thresholds to specify. Review the options below. Click **Next** when finished.

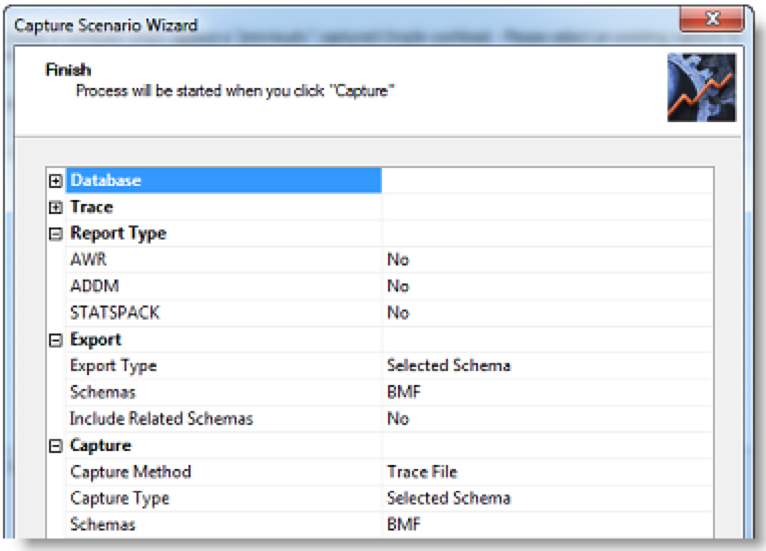
Stop capture if CPU percentage exceeds	Enter a percentage. If CPU usage exceeds this value, Benchmark Factory stops the capture process.
Stop capture if disk/tablespace free space falls below	Enter a value for free space in MB. If the amount of free space falls below this level, Benchmark Factory stops the capture process.

17. **Capture Control.** Use the Capture Control page to specify when to start the capture.

    You can start the capture immediately after finishing the wizard, or you can schedule the capture. Review the options below. Click **Next** when finished.

Start Capture/Export	**Immediately**—Select to start the capture process immediately. **Scheduled (via database schedule)**—Select to schedule the capture process. Then specify a date and time based on the server time.
Capture Automatically	**Stop After**—Use the sliders to specify a duration for the capture process. The default is 30 minutes.

18. **Start the capture.** The Finish page provides capture specifications. To start the capture (or enable the schedule), click **Capture**.



Capture Scenario Wizard

Finish

Process will be started when you click "Capture"

⊞ Database	
⊞ Trace	
⊟ Report Type	
AWR	No
ADDM	No
STATSPACK	No
⊟ Export	
Export Type	Selected Schema
Schemas	BMF
Include Related Schemas	No
⊟ Capture	
Capture Method	Trace File
Capture Type	Selected Schema
Schemas	BMF

19. **Capture Status.** The Capture Status window opens providing export or capture status and process details.

    Immediate capture: If you selected to start the capture immediately, the Capture Status window displays information about the capture process, such as the status of the export process and the number of sessions captured.

    - Click the **Click here for details** link to view more-detailed information during the export or capture. You can view which objects are exporting or which session/user is currently being captured and the total number of sessions captured.
    - Click **Settings** to review the capture description.

    Scheduled capture: If you scheduled the capture process, leave the capture wizard open to run the capture at the scheduled time.

20. **Transfer and process files.** When the workload capture is finished, the Oracle capture files are processed.

    - If you stopped the capture before it was finished, Benchmark Factory prompts you to process the files.
    - If you specified a client-side directory (instead of a shared directory) as the *Capture Directory* (directory from which the client replays the capture), BMF prompts you to transfer the capture files to your client-side directory now. Copy or move the trace/xml and export (DMP) files from the directory where they were generated (*Database Server Directory*) to the client-side *Capture Directory*. Place the files into the existing capture-named sub-directory. After transferring the files, click **OK** in the message box. BMF then processes the necessary files and updates the project file.

      Optionally, if you want to transfer the files at a later time, click **Cancel**.

21. Click the **Click here for details** links to review the export and capture processes. Close the Capture Status window when finished. You can open the Capture Status window again from the Captures tab in the Benchmark Factory console. In the Captures tab, right-click a capture and select **View Status**.

22. You can now replay the workload.

To replay an Oracle workload

1. Select **File | New** to open the New Job Wizard.
2. On the **Database Under Test** page, select the database to test.
3. Then click Add Workload.

4. On the Workload page, select **Capture/Replay Test** from the drop-down list.
5. Then select the **Capture and Replay Oracle Workload** option.
6. Click Select Test.
7. **Select a capture.** On the Replay Workload page, select a capture from the **Capture to Replay** field using one of the following methods:

    - Click the browse button and navigate to the directory where the capture files are located. Select the project (.mse) file for the capture you want to replay.
    - Select one of your previously-replayed captures from the drop-down list.

    Benchmark Factory loads the capture details into Capture Properties fields.

8. **Transfer and process files.** If you did not transfer the trace or .xml capture files to the client-side directory at the time of the capture, or you transferred the files at a later time, Benchmark Factory warns you that the capture data must be processed.

    - To process the files you transferred after the capture, click **Yes** in the warning message. Benchmark Factory closes the New Job Wizard and begins processing the capture data.
    - To transfer the files now, first click **Yes** in the warning message. Benchmark Factory then prompts you to transfer the files. Transfer the files, and then click **OK** in the prompt.

    When the process is finished, close the Capture Status window. Then open the New Job Wizard again and select the project (.mse) file again.

9. **Import Test.** Click Import Test. The test is added to the workload and the test Summary page displays.
10. If you exported the database objects and data required to replay the workload, Benchmark Factory prompts you for the location of this file. Enter the path to the file location on the database server (or network).
11. On the Summary page, you can click each of the links to go directly to an option to modify it.
12. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

13. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Considerations When Performing an Oracle Capture and Replay

Consideration	Description
Trace file capture and replay times (AWR) might not match	When using trace files as the capture method, you could find that the replay time as reported by the AWR does not match your specified capture time. Benchmark Factory trace method only captures sessions that are initiated after the capture starts. If there is a lag between when the capture starts and when the first session is captured, this lag is reflected by the mismatched times.
Oracle Enterprise Manager TPS results	If you choose to collect database wide performance statistics from the AWR, the TPS results from Benchmark Factory and Oracle Enterprise Manager may not match. This is because Benchmark Factory displays TPS for the workload, while OEM displays TPS for *all* the work the server is doing.

Notes:

- Database objects created by Benchmark Factory for the capture process are automatically dropped after the capture is finished.
- You can manage your existing capture projects from the **Captures** tab in the Benchmark Factory console.

Capture and Replay SQL Server Workload

Benchmark Factory allows you to capture SQL Server workloads, then replay those workloads to view real-time transactions and response results.

In Benchmark Factory, use one of the following workflows to capture a SQL Server workload:

- **New Job Wizard**—Use the New Job Wizard (**File | New**) to guide you through the steps of setting up and capturing the workload. Then add the captured workload to your new job to run it immediately or at a later time.
- **Capture Scenario Wizard**—Select **Tools | Capture Workload** to open the Capture Scenario Wizard.

Note: This feature is not available in the freeware edition of Benchmark Factory.

To capture a SQL Server workload

1. Select **File | New** to open the New Job Wizard.
2. On the **Database Under Test** page, select the database to test.
3. Then click .

4. On the Workload page, select **Capture/Replay Test** from the drop-down list.
5. Then select the **Capture and Replay SQL Server Workload** option.
6. Click .

7. On the Replay Workload page, click **Perform New Capture**. The Capture Scenario Wizard opens and the New Job Wizard closes.

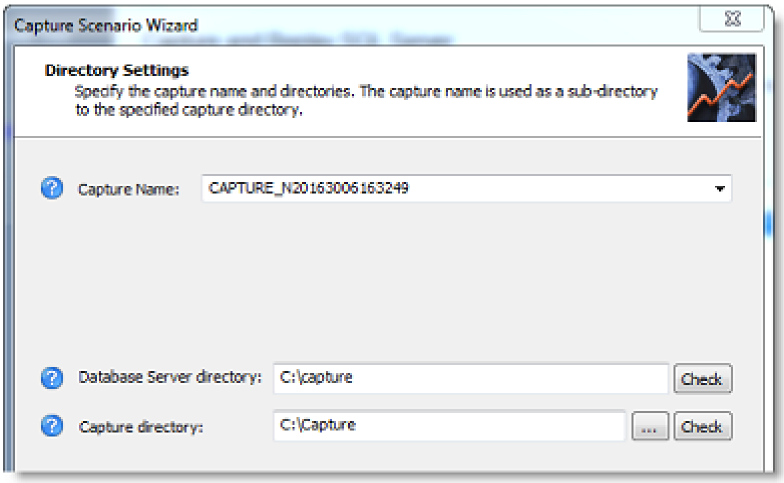
   Tip: You can also create a new capture using the same settings as an existing capture project. In the Captures tab of the Benchmark Factory console, right-click a capture and select **Repeat Capture**.

8. **Capture Scenario Wizard.** In the Capture Scenario Wizard, enter connection information for the database from which you want to capture a workload. See Create SQL Server Connection on page 48 for more information. Click **Next** when finished.

9. **Directory Settings.** On the Directory Settings page, specify a SQL Server server-side directory in which to place the capture files. Also, specify the capture directory from which Benchmark Factory will replay the files. Review the following for additional information:

   Click **Next** when finished.

Capture Name	Enter a name for the capture, or use the default. This name is used for the sub-directory where the capture files are stored in the Capture Directory you specify. ℹ **Note:** Only alpha and numeric characters and the underscore (_) are permitted. The name must begin with a letter.
Database Server Directory	Specify a server-side directory where Benchmark Factory should place the capture files. Specify the path as the server sees it. ℹ **Note:** You can specify a network directory here. Enter the full network path to the network directory. The database service must be able to access the network directory.
Capture Directory	Specify a directory where Benchmark Factory will look for the capture files to replay. Do one of the following: • **Manual transfer:** Specify a client-side directory and then manually transfer the capture files to this location. • **Shared directory:** Specify the same server-side (or network) directory you specified in the preceding field. However, enter the path as the client computer sees it. To use this method, you must first map a drive on the client computer to the Database Server Directory (server-side or network directory). Then, browse to and select that mapped drive. **Selecting a shared directory:** The shared capture directory can be on the database server as a local directory or on a network file server as a local directory. Either location must be accessible by the Benchmark Factory client computer (Windows network share, Samba, or NFS). A network file server is the preferred location for the following reasons: • This minimizes total network I/O required for capture and replay. • The shared directory can be placed on an I/O subsystem with sufficient I/O bandwidth to handle both the concurrent I/O writing and the cumulative size of the trace and export files.



Capture Scenario Wizard

Directory Settings

Specify the capture name and directories. The capture name is used as a sub-directory to the specified capture directory.

Capture Name: CAPTURE_N20163006163249

Database Server directory: C:\capture

Capture directory: C:\Capture

10. **Capture Scope.** In the Capture Scope page, select one or more databases from which to capture activity. Review the options below. Click **Next** when finished.

Capture activity for entire database	Select to capture all activity for the entire database.
Capture only activity for selected databases	Select to capture activity only for the selected databases. Then select the databases from which to capture activity.

11. **Backup Scope.** Benchmark Factory can export/backup the database objects and data required to replay the workload on the same data snapshot. Review the description below. Click **Next** when finished.

Perform backup as part of the Capture Process	Select this option to instruct Benchmark Factory to export/backup the databases selected on the previous page (Capture Scope). The backup is performed during the capture procedure. **Note:** If you selected to capture activity for the entire database, exporting the entire database can require significant time and space.

12. **Filter Settings.** (Optional) The Filtered Settings page allows you to add filters to exclude activity from the capture.

    - To add a filter, click **Add**. Then specify parameters for the filter. Click **Next** when finished.

13. **Capture Thresholds.** Benchmark Factory allows you to specify limits for CPU usage and free space during a capture. If levels exceed the values you specify, the capture process is stopped.

    Benchmark Factory displays the current values to help you determine the best thresholds to specify. Review the options below. Click **Next** when finished.

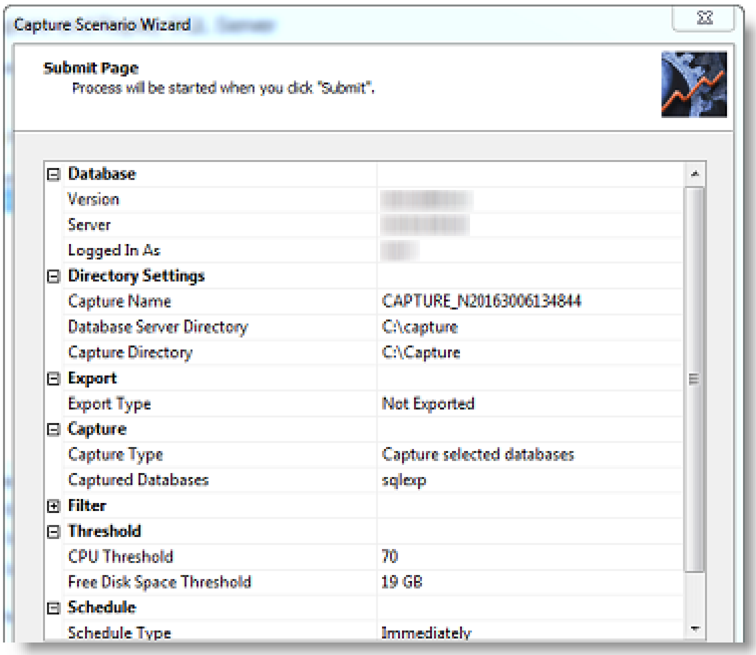
Stop capture if ...	Description
host CPU percentage of database server exceeds	Enter a percentage. If CPU usage exceeds this value, Benchmark Factory stops the capture process.
trace file disk free space falls below	Enter a value for free space in GB. If the amount of free space falls below this level, Benchmark Factory stops the capture process.

Note: The Capture Thresholds feature is not available for SQL Server running in a Linux environment.

14. **Schedule Job.** You can schedule a capture or start it immediately. Review the options below. Click **Next** when finished.

Start Capture/Export	**Immediately**—Select to start the capture process immediately. **Later**—Select to schedule the capture process. Then specify a date and time based on the server time.
Capture Automatically	**Stop After**—Use the sliders to specify a duration for the capture process. The default is 30 minutes.

15. **Start the capture.** The Submit page provides capture specifications. To start the capture (or enable the schedule), click **Submit**.



Capture Scenario Wizard [redacted]

Submit Page

Process will be started when you click "Submit".

⊟ **Database**	
Version	[redacted]
Server	[redacted]
Logged In As	[redacted]
⊟ **Directory Settings**	
Capture Name	CAPTURE_N20163006134844
Database Server Directory	C:\capture
Capture Directory	C:\Capture
⊟ **Export**	
Export Type	Not Exported
⊟ **Capture**	
Capture Type	Capture selected databases
Captured Databases	sqlexp
⊞ **Filter**	
⊟ **Threshold**	
CPU Threshold	70
Free Disk Space Threshold	19 GB
⊟ **Schedule**	
Schedule Type	Immediately

16. **Capture Status.** The Capture Status window opens providing export and/or capture status and details of the process.

    Immediate capture: If you selected to start the capture immediately, the Capture Status window displays information about the capture process, such as the status of the export process and the number of sessions captured.

    - Click the **Click for capture details** link to view more-detailed information during the export or capture. You can view which objects are exporting or which session/user is currently being captured and the total number of sessions captured.

    - Click **Properties** to review the capture description.

17. Click the **Click for capture details** links to review the export and capture processes. Close the Capture Status window when finished. You can open the Capture Status window again from the Captures tab in the Benchmark Factory console. In the Captures tab, right-click a capture and select **View Status**.

To replay a SQL Server workload

1. Select **File | New** to open the New Job Wizard.
2. On the **Database Under Test** page, select the database to test.
3. Then click Add Workload.
4. On the Workload page, select **Capture/Replay Test** from the drop-down list.
5. Then select the **Capture and Replay SQL Server Workload** option.
6. Click Select Test.
7. **Select a capture.** On the Replay Workload page, select a capture from the **Capture to Replay** field using one of the following methods:

    - Click the browse button and navigate to the directory where the capture files are located. Select the project (.cpj) file for the capture you want to replay.

    - Select one of your previously-replayed captures from the drop-down list.

    Benchmark Factory loads the capture details into Capture Properties fields.

8. **Import Test.** Click

Import Test. The test is added to the workload and the test Summary page displays.

9. On the Summary page, you can click each of the links to go directly to an option to modify it.

10. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

11. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Notes:

- Database objects created by Benchmark Factory for the capture process are automatically dropped after the capture is finished.
- You can manage your existing capture projects from the **Captures** tab in the Benchmark Factory console.

Run the Capture Scenario Wizard

You can run the Capture Scenario Wizard as a standalone application to capture an Oracle or SQL Server workload. Then replay the workload in Benchmark Factory. This utility provides an easy way for you to capture your database workloads without requiring the installation of Benchmark Factory.

After a workload is captured, the capture files can be distributed to others to replay and test in Benchmark Factory.

This utility is the same Capture Scenario Wizard that opens from the New Job Wizard in Benchmark Factory and it produces the same type of capture and export files.

Prerequisites

Oracle or SQL Server Capture. The following is required to run the Capture Scenario Wizard to capture any workload:

- Microsoft MSXML must be installed to run the Capture Scenario Wizard.

Oracle Capture. The following is required to run the Capture Scenario Wizard to capture an Oracle workload:

- An Oracle Client (version 9 or later) must be installed to create an Oracle connection and capture the workload.
    - The 32-bit version of the Capture Wizard requires the 32-bit Oracle Client.
    - The 64-bit version of the Capture Wizard requires the 64-bit Oracle Client.

Run the Capture Scenario Wizard

To run the Capture Scenario Wizard standalone application

1. Navigate to the Benchmark Factory application directory where Benchmark Factory is installed and open the **bin** directory.

   For example, C:\Program Files (x86)\Quest Software\Benchmark Factory 7.5.0\bin

2. In the **bin** directory, double-click **BFCapture.exe** to start the application.
3. Select the capture type (Oracle or SQL Server).
   - To capture an Oracle workload, see Capture and Replay Oracle Workload and skip ahead to the Capture Scenario Wizard step.
   - To capture a SQL Server workload, see Capture and Replay SQL Server Workload and skip ahead to the Capture Scenario Wizard step.

To start the Capture Scenario Wizard from the command line, see Run the Capture Wizard from the Command Line.

Run the Capture Wizard from the Command Line

You can use the command line to start and run the Capture Scenario Wizard.

When running the Capture Scenario Wizard from the command line, you can specify the following parameters:

Parameter	Description
-?	Displays Help

-O | -S	Specifies the database type for the capture.
	-O performs an Oracle capture.
	`BFCapture.exe -O`
	-S performs a Microsoft SQL Server capture.
	`BFCapture.exe -S`
	These parameters are ignored when used with another parameter.
	If you do not specify a capture database type (Oracle or SQL Server), the Capture Scenario Wizard prompts you to select one.

-V	View the status of the capture for the capture project file specified (*.mse or *.cpj). `BFCapture.exe -V <capture project file>`
-D	Deletes a capture. Specify the capture project file to delete (*.mse or *.cpj). Deletes the project file and the capture files. You cannot delete a running capture. `BFCapture.exe -D <capture project file>`
-C	Starts a new capture using the values in the specified capture project file (*.mse or *.cpj). Does not overwrite the specified capture. `BFCapture.exe -C <capture project file>`

Note: The parameters -V, -D, and -C cannot be used at the same time and must have a capture project file location specified.

Replay Load from an Oracle Trace File

Benchmark Factory allows you to import the SQL contained within an Oracle trace file. The resulting workload contains a user scenario consisting of the timed event sequence of captured SQL transactions.

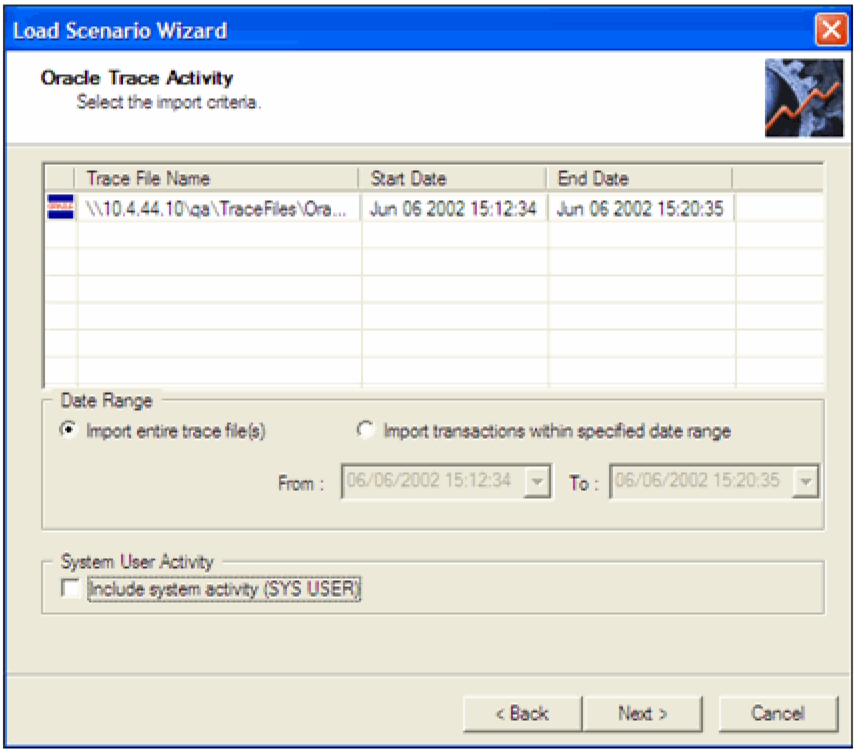
Importing an Oracle Trace file allows you to replay your database activity with the same timing as was originally captured on an existing in-house Oracle database application.

Note: To create an Oracle trace file which includes bind parameters, a level number of 4 must be used. For example, to start an Oracle trace for a user session, use the following command:

```
ALTER SESSION SET EVENTS '10046 trace name context forever, level 4'
```

To import an Oracle trace file

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Capture / Replay Test* from the drop-down list.
3. Select **Capture load from an Oracle trace file**.
4. Click the **Select Test** button. The Oracle Trace Input dialog opens.
5. Click **Add Trace** and browse to and select the trace file (or files).
   - To add additional files, click **Add Trace**.
   - To remove a file from the list, select the file and click **Remove Trace**.
6. When you finish inputting files, click **Next**. The Oracle Trace Activity dialog opens.



Load Scenario Wizard

Oracle Trace Activity

Select the import criteria.

Trace File Name	Start Date	End Date
\\10.4.44.10\qa\TraceFiles\Ora...	Jun 06 2002 15:12:34	Jun 06 2002 15:20:35

Date Range

● Import entire trace file(s) ○ Import transactions within specified date range

From : 06/06/2002 15:12:34 To : 06/06/2002 15:20:35

System User Activity

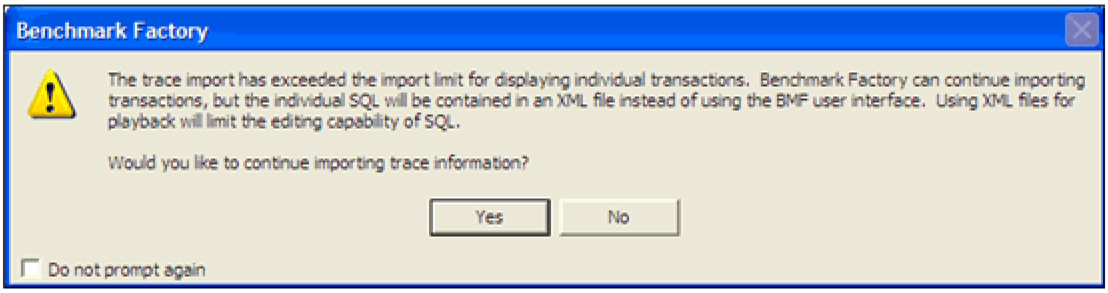
☐ Include system activity (SYS USER)

7. Select import criteria.
   - **Import entire trace file**—Imports all activity in the trace file or files.
   - **Import transactions within specified date range**—Select this option and enter a date range to import only the activity within the date range.

- **Include system activity**—Select to include system activity (SYS USER).

  Note: It is **not** recommended to include system activity (SYS USER). This imports Oracle SQL statements routinely run by Oracle for maintenance reasons and is not required for the load testing process.

8. Click **Finish**.
9. If the trace import file exceeds the Benchmark Factory limit for displaying individual transaction, the following dialog displays:



Benchmark Factory

The trace import has exceeded the import limit for displaying individual transactions. Benchmark Factory can continue importing transactions, but the individual SQL will be contained in an XML file instead of using the BMF user interface. Using XML files for playback will limit the editing capability of SQL.

Would you like to continue importing trace information?

□ Do not prompt again

If you click **Yes**, the trace file import continues. Individual SQL is converted to .xml files. You can then edited the .xml files in the Benchmark Factory Session Editor.

10. The test is added to the workload and the test Summary page displays.
11. On the Summary page, you can click each of the links to go directly to an option to modify it.
12. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

13. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Replay from SQL Server Trace Table

Benchmark Factory allows you to import SQL from a SQL Server trace table. This allows you to replay your database activity with the same timing as was originally captured on an existing in-house SQL Server database application.

The resulting workload contains a user scenario consisting of the sequence of captured SQL transactions.

Before importing SQL from a SQL Server trace table, use SQL Profiler to save the SQL trace as a trace table. See the following topics for more information:

- Create a SQL 2000 Trace Table Using the SQL Server Profiler
- Create a SQL 2005 Trace Table Using the SQL Server Profiler
- Creating a SQL 2008/2008 R2 Trace Table Using the SQL Server Profiler

To import a workload from a SQL Server trace table

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Capture / Replay Test* from the drop-down list.
3. Select **Replay load from SQL Server trace table**.
4. Click the **Import Test** button.
5. In the SQL Server Authentication dialog, enter the connection information for the database containing the trace table to import.
6. Click **Next**. The SQL Server Trace Table dialog opens.
7. Select a database from the **Select a Database** drop-down list.
8. Select a table from the **Select a Table** drop-down list.
9. Click **Next**. The SQL Server Trace Processing dialog displays.
10. Click **Finish**. The test is added to the workload and the test Summary page displays.
11. On the Summary page, click each link to go directly to that option to modify it.
12. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

13. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Replay Load from an ODBC Trace File

Benchmark Factory allows you to import SQL from an ODBC trace file generated by the ODBC Data Source Administrator. This allows you to load test your database using workloads from an existing in-house ODBC database application.

The resulting workload contains a user scenario consisting of the ordered sequence of captured SQL Transactions from the ODBC trace.

See Creating an ODBC Trace File for instructions on how to create an ODBC trace file.

To import an ODBC trace file

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Capture / Replay Test* from the drop-down list.
3. Select **Replay load from an ODBC trace file**.
4. Click the **Import Test** button.
5. Browse to and select the ODBC trace (log) file to import.

6. Click **Next**. The ODBC Trace Processing dialog opens. When the processing is finished, the number and types of SQL extracted from the file displays.

7. Click Finish. The test is added to the workload and the test Summary page displays.

8. On the Summary page, click each link to go directly to that option to modify it.

9. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

10. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Import from a Delimited Text File

Benchmark Factory allows you to create a workload (or add to a workload) by importing SQL from a delimited text file. This allows you to load test your database using workloads from existing in-house generic database applications.

To Import from a Delimited Text File

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.

2. On the workload page, select *Capture / Replay Test* from the drop-down list.

3. Select **Import from a Delimited Text File**.

4. Click the **Import Test** button.

5. In the Text File Input dialog, browse to and select a file. Click **Next**.

6. In the Select File Format page, select the desired format (*Delimited* or *Fixed field*).

   - If you selected *Delimited*, select the desired row delimiter, text qualifier, and rows to skip.
   - If you selected *Fixed Field*, select the desired row delimiter and rows to skip.

7. Click **Next**.

8. If you selected *Delimited*, the Specify Column Delimiter page opens. Select delimiter options.

   Note: Only the first 20 rows of the file display.

9. If you selected *Fixed field*, the Fixed Field Column Positions page opens. Configure the column or columns.

   Note: Only the first 20 rows of the file display.

10. Click **Finish**.The test is added to the workload and the test Summary page displays.

11. On the Summary page, click each link to go directly to that option to modify it.

12. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the

following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

13. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Import from Oracle Dynamic Performance View

Benchmark Factory allows you to import SQL from Oracle Dynamic Performance Views. This allows you to import SQL transactions identified by the Oracle for further testing under a load.

The resulting workload is a mixed workload containing either the most-often executed, the most time-consuming, or the most-recently executed SQL transactions.

To import from the Oracle Dynamic Performance Views

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Capture / Replay Test* from the drop-down list.
3. Select **Import from Oracle Dynamic Performance View**.
4. Click the **Import Test** button. The Oracle DPV Connection dialog opens.
5. Enter the connection information and click **Next**. The Oracle DPV Import dialog opens.
6. Select the type of SQL statements to import.
7. Select the maximum number of SQL statements to return.
8. Deselect the **Include system activity (SYS USER)** option to exclude system activity.

   Note: It is not recommended to include system activity. Including it imports Oracle SQL statements routinely run by Oracle for maintenance reasons, and is not required for the load testing process.

9. Click **Finish**. The test is added to the workload and the test Summary page displays.
10. On the Summary page, click each link to go directly to that option to modify it.
11. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Specify sampling, pre-sampling, and user start-up times for the test. See

Tab	Description
	Timing Tab on page 100 for more information.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

12. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Import from Quest Tools Export File

You can create a workload (or add to an existing workload) by importing a file exported by another Quest Software tool.

To import a file exported by another Quest Software tool

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Capture / Replay Test* from the drop-down list.
3. Then select **Import from Quest Tools Export File**.
4. Click the **Import Test** button.
5. Browse to and select a file to import. Click **Finish**.
6. The test is added to the workload and the test Summary page displays.
7. On the Summary page, you can click each of the links to go directly to an option to modify it.
8. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Specify sampling, pre-sampling, and user start-up times for the test. See Timing Tab on page 100 for more information.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

9. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Artificial Test for Desired Effect

Artificial Test for Desired Effect

Benchmark Factory allows you to design a workload based on a mixture of one of the following:

- Industry standard benchmarks
- Database application types (transaction/process types)
- Database transactions (SQL statement types)
- Hardware stressing focal points (database resources)

To create an artificial test for desired effect

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Artificial Test for Desired Effect* from the drop-down list.
3. Select one of the following test types:
    - **Based on user selection of one or more industry standard benchmarks**—Design a workload by specifying the percentages of industry standard benchmark (TPC-C, TPC-H, TPC-E, etc.) to use for the workload.
    - **Based on user selection of one or more database application mixtures**—Design a workload based on a mixture of OLTP and Data Warehouse type transactions, specified as percentages.
    - **Based on user selection of one or more database transactional mixtures**—Design a workload based on a mixture of SELECT, INSERT, UPDATE, and DELETE SQL statements, specified as percentages.
    - **Based on user selection of one or more hardware stressing focal points**—Design a workload by specifying the percentages of transactions that stress CPU, Memory, Disk IO, and Network.
4. Click the **Select Test** button at the bottom of the page.
5. Specify a weight and benchmark scale factor for each benchmark, transaction type, SQL type, or database resource.
6. Click **Add Test**.
7. The tests are added to the workload and the workload Summary page opens.
    - On the Summary page, you can click each of the links to go directly to an option to modify it.
8. You can also select the Test Options tab. Each test/step in the workload may have a different set of options available in its Test Options tab. See the following for more information.
    - Transactions Tab
    - Specify User Load
    - Timing Tab
    - Advanced Tab
    - Benchmark Scale Factor
    - Options Tab (Create Objects Step)
    - Customize SQL for Creating Objects
9. After specifying options for this test, you can add another test to the job, configure job setup options, save

and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Scalability Tests

Create SQL Scalability Test

The SQL Scalability test allows you to execute SQL statements, allowing users to spot potential issues not seen with a single execution. Users can run variations of a SQL statement in order to find the SQL that will perform the best under a load test.

Use this procedure to create a new *SQL Scalability* test or a *Custom Scalability* test.

To create a SQL Scalability Test

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, do one of the following:
   - Select *Scalability Test* from the drop-down list. Then select the **Test SQL for Scalability** option.
   - Select *Custom Test*. Then select the **Scalability test compares ...** option.
3. Click the **Add Test** button. The test is added to the workload and the test Summary page displays.
4. On the Summary page, click each link to modify the specific test option.
5. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Add transactions for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Select one of the following options: ● **Number of executions per iteration**-Each transaction is executed by each user for a specified number of times (recommended). ● **Execution time per iteration**-Executes each transaction for the specified length of time.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

6. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Test Stored Procedures for Scalability

Stored procedures enables developers who write stored procedures in Oracle and SQL Server to test performance and scalability of the logic under varying user load levels. Benchmark Factory allows you to import

SQL from database stored procedures into a Mix Load Scenario (workload). This gives you the ability to import SQL transactions, from stored procedures, for further testing under load.

To test Stored Procedures for Scalability

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Scalability Test* from the drop-down list.
3. Select the **Test Stored Procedures for Scalability** option.
4. Click the **Select Test** button.
5. In the Select Stored Procedures dialog, select the procedures from which to import SQL.
6. Click **Next**.
7. In the Bind Parameters page, select a SQL statement and enter a bind variable if necessary.
8. Click **Finish.** The SQL is imported and the Summary page displays.
9. Click the links on the Summary page to modify test and job setup options.
10. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Specify sampling, pre-sampling, and user start-up times for the test. See Timing Tab on page 100 for more information.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

11. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Custom Tests

Custom Tests

Custom tests allow you to create workloads from user provided SQL. These tests provide flexibility for your load testing requirements. The following customs load scenarios are provided:

- Mix Test—Runs a transaction mix based upon weights for a specified time at each predetermined user load level.
- Replay Test—Runs multiple transactions with each one running independently on a specified number of users.
- Goal Test—Used to find maximum throughput or response time values. A transaction mix is executed at a range of user load levels.

- Scalability Test—Compares the performance of SQL Statement variations under a workload. Each transaction will execute individually for each user load and timing period.

Create Mix Test

A Mixed Workload test runs for a specified time at each predetermined user load level. Each user will run a transaction mix based upon the weights defined on the transactions tab. For example, if a test has two transactions, A and B, with A having a weight of one and B having a weight of four, on average B will run four times for every time A is run once. The run order will be randomly generated for each user so they are not all running the same transaction simultaneously. That run order is used for that user each time the test is performed to ensure reproducible results.

To Create a Custom Mix Test

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Custom Test* from the drop-down list.
3. Select the **Mix test executes ...** option.
4. Click the **Add Test** button.The test is added to the workload and the test Summary page displays.
5. On the Summary page, click the link beside **Custom Mix Load Scenario** to add transactions. See Transactions Tab on page 98 for more information.
6. Then specify the weight for each transaction.
7. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Specify sampling, pre-sampling, and user start-up times for the test. See Timing Tab on page 100 for more information.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

8. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Create Replay Test

A Replay Test runs multiple transactions with each one running independently on a specified number of users. The test will run until the defined number of executions for each transaction or a specified time limit is reached.

To create a Custom Replay Test

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Custom Test* from the drop-down list.
3. Select the **Replay test executes ...** option.

4. Click the **Add Test** button. The test is added to the workload and the test Summary page displays.

5. On the Summary page, click the link beside **Custom Replay Load Scenario** to add transactions or a user scenario. See Transactions Tab on page 98 for more information.

6. Then select the number of users and/or number of executions for each transaction or user scenario.

7. To jump to the Test Options tab, click **Test Options** at the bottom of the Summary page. Review the following for more information:

Tab	Description
Transactions tab	Modify the transactions and the transaction mix for the test. For a custom test, add transactions. See Transactions Tab on page 98 for more information.
Options tab	Enable scaling for the user scenario. See Options Tab (Capture/Replay) on page 103 for more information.
Advanced tab	Specify Repository options, error handling, and connect/disconnect options for the test. See Advanced Tab on page 101 for more information.

8. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Create Goal Test

A Goal Test uses a transaction mix based on userload intervals. The userloads are determined by setting a beginning, ending, and interval value. This value specifies an interval to increase the number of users tested for each iteration of the Goal Test. When run, the specified goal criterion is evaluated at the end of each iteration and the test ends once the goal or maxim user load has been reached.

To create a goal test

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.

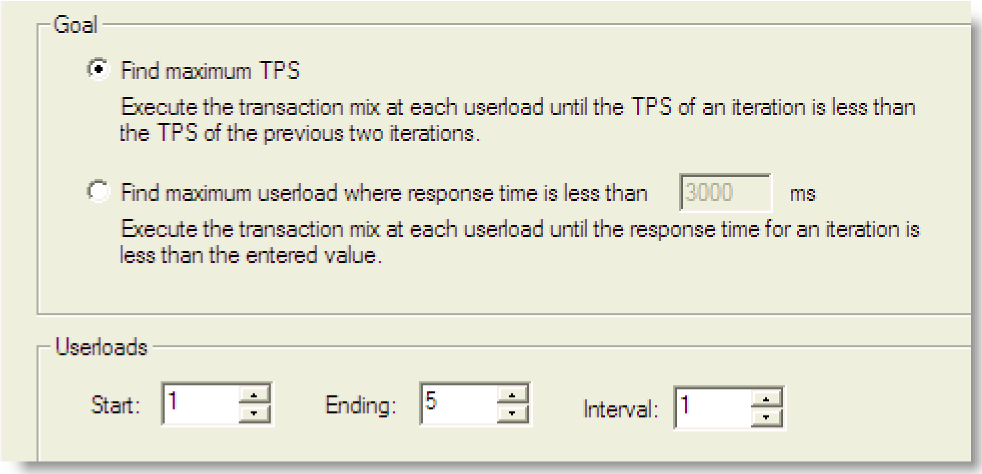
2. On the workload page, select *Custom Test* from the drop-down list.

3. Select the **Goal test is used ...** option.

4. Click the **Add Test** button.The test is added to the workload and the test Summary page displays.

5. On the Summary page, click the link beside **Max TPS ...** to add transactions. See Transactions Tab on page 98 for more information.

6. To change the goal, select the Options tab (under Test Options tab).



Goal

● Find maximum TPS

Execute the transaction mix at each userload until the TPS of an iteration is less than the TPS of the previous two iterations.

○ Find maximum userload where response time is less than 3000 ms

Execute the transaction mix at each userload until the response time for an iteration is less than the entered value.

Userloads

Start: 1 Ending: 5 Interval: 1

7. Select one of the following:

   - **Find maximum TPS**—Selecting this option will show how the maximum TPS can be reached during a load test.

   - **Find maximum userload where response time is less than <*n*> ms**—Selecting this option shows how a maximum userload can be reached when response time is less than the specified value.

8. To modify the userload range and interval, select the Options tab (under Test Options tab). Specify a range and then an interval at which to increase users after each iteration.

   - For this test, the default values are: a range of 1-5 and an interval of 1. When run, User Loads of 1, 2, 3, 4, 5 are run until the TPS of one run is less than the TPS of the previous run.

9. Specify more options for this test by reviewing the Timing and Advanced tabs. Review the following for additional information:

   - **Timing tab**—Specify sampling, pre-sampling, and user start-up times for the test. See Timing Tab on page 100 for more information.

   - **Advanced tab**—Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

10. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Create SQL Scalability Test

The SQL Scalability test allows you to execute SQL statements, allowing users to spot potential issues not seen with a single execution. Users can run variations of a SQL statement in order to find the SQL that will perform the best under a load test.

Use this procedure to create a new *SQL Scalability* test or a *Custom Scalability* test.

To create a SQL Scalability Test

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.

2. On the workload page, do one of the following:

   - Select *Scalability Test* from the drop-down list. Then select the **Test SQL for Scalability** option.

   - Select *Custom Test*. Then select the **Scalability test compares ...** option.

3. Click the **Add Test** button. The test is added to the workload and the test Summary page displays.

4. On the Summary page, click each link to modify the specific test option.

5. Or you can select the Test Options tab to modify all test options. Review the following for additional information:

Tab	Description
Transactions tab	Add transactions for the test. See Transactions Tab on page 98 for more information.
User Load tab	Modify the number of users per test iteration. See Specify User Load on page 134 for more information.
Timing tab	Select one of the following options: - **Number of executions per iteration**-Each transaction is executed by each user for a specified number of times (recommended). - **Execution time per iteration**-Executes each transaction for the specified length of time.
Advanced tab	Specify Repository options, error handling, and database checkpoints for the test. You can also specify a file to execute at the beginning or end of each iteration. See Advanced Tab on page 101 for more information.

6. After specifying options for this test, you can add another test to the job, configure job setup options, save and close the job, run the job, or schedule the job. For more information about each of these steps, see Quickstart: Create a New Job.

Add a Create/Delete Benchmark Objects Test

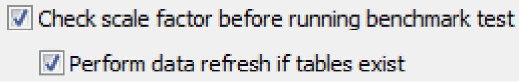
The Create/Delete Benchmark Objects step creates or clears the database objects (tables/indexes, etc.) needed to run industry standard benchmarks. The step is included when you create an Industry Standard Benchmark Test. You can also add the step to an existing workload through the Edit Job Wizard.

Note: To create, delete, and modify database tables/indexes you **must** have the user login rights required.

To add a Create/Delete Benchmark Objects Test to a Workload

1. Use one of the following methods to open the Job Wizard:
   - Select **File | New** to open the New Job Wizard.
   - Select an existing job and click **Edit Job**.
2. Click **Workload** in the left pane. If this is an existing job, click **Add Another Test/Step**
3. On the Workload page, select **Create/Delete Benchmark Objects Test** from the drop-down list.
4. Then select the benchmark type from the **Benchmark Type** drop-down list.
5. Select one of the following options:
   - **Create Benchmark Objects**—Creates and initializes benchmark objects (tables, indexes, etc.).
   - **Delete Benchmark Objects**—Deletes existing benchmark objects (tables, indexes, etc.).

6. If you selected **Create Benchmark Objects**, select one of the following options:

Create/Load objects if objects don't exist (no backup sets created)	If objects do not exist, Benchmark Factory creates the objects and loads data. If you select this option, Benchmark Factory does not create backup tables.
Delete benchmark objects after completion of job	Benchmark Factory deletes objects after a job is finished.
Refresh data using backup sets	When backup sets exist, Benchmark Factory reloads data and recreates indexes. If backup sets do not exist, Benchmark Factory creates objects and loads data. Then Benchmark Factory creates the backup tables.
Recreate objects and reload data every execution	Benchmark Factory always deletes the existing objects and then recreates/reloads for each job execution. This allows you to ensure that the dataset is always in initialized status.
Refresh data using inserts	If data does not exist, Benchmark Factory uses bulk insert to load data. If the following Benchmark Settings are selected,  ☑ Check scale factor before running benchmark test ☑ Perform data refresh if tables exist and the Benchmark Scale factor of the existing tables does not match the new requirement, the process of inserting additional data or deleting data my run long, depending on the size of the existing data and number of indexes. **Note:** If database does not support bulk insert, insert is used.

7. Click the **Import Test** button at the bottom of the page.

8. **Summary.** The test is added to the workload and the Summary page opens. Review the following editable options for this test/step.

   - **Scale**—Click the **Scale** link to modify the scale factor. The Scale page opens. In the **Benchmark Scale** field, select a new scale factor. See Benchmark Scale Factor on page 134 for more information.

   Note: The test is added to the top of the workload list. To move the test down (or up) in the list, right-click the test and select **Move Down** (or **Move Up**).

9. **Test Options.** To modify test options for the Create Objects test/step, select the Test Options tab. Review the following for more information:

   - **Scale**—To modify the scale factor. See Benchmark Scale Factor on page 134 for more information.

   - **Options**—To modify options for object creation and retention. See Options Tab (Create Objects Step) on page 95 for more information.

   - **Custom Creation SQL**—To use custom SQL to create objects. See Customize SQL for Creating Objects on page 96 for more information.

Tip: You can specify a default setting for Object/Data Retention in **Edit | Settings | Benchmarks**.

Execute External File

You might want to add an executable file (.bat or .exe) as part of a job. This gives you the flexibility to set up your database before running a workload. For example, you could run a .bat file to set up a prepared SQL statement before running a workload.

To add an Execute File step to a workload

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Execute External File* from the drop-down list.
3. Enter the path or browse to and select a file to execute.
4. To add a timeout, select the **Enforce Timeout** option and enter the number of seconds for Benchmark Factory to wait for the execution to complete before continuing the job.

   You can specify a default setting for this option in **Edit | Settings | Execute File**.

Test Options - Create Objects

Benchmark Scale Factor

When you create an Industry Standard Benchmark test or add a Create Benchmark Objects step, Benchmark Factory allows you to adjust the Benchmark scale factor. The scale factor increases the size of a database during the testing process allowing realistic testing.

By increasing the scale factor, the number of rows added to the tables increases, allowing for the use of larger user loads. This places a greater stress on the system-under-test. During load testing, you can run various tests with different Benchmark Scale factors to ensure your database performs at varying user-load levels.

You can adjust the Benchmark Scale factor when creating a new Industry Standard Benchmark Test or when adding a Create Benchmark Objects step.

To change the Benchmark Scale factor

1. In the New Job Wizard or Edit Job Wizard, select the Create Objects step (in the Workload) for which you want to adjust the scale.
2. Select the **Test Options** tab and then the **Scale** tab.
3. In the **Benchmark Scale** field, specify a scale factor.

   Valid benchmark scales are dependent upon the benchmark selected. For example, the AS3AP benchmark scales by a factor of 10 with valid scales of "1, 10, 20, 30, etc." The TPC-C benchmark scales by a factory of 1, with valid scales of "1, 2, 3, 4, etc." Refer to the specific benchmark standard for more information.

   (Replication test only) For a Replication test, each increment in scale factor adds 1000 rows to a table.
4. (Optional) Instead of specifying a scale factor, you can specify a **Benchmark Size**. The Benchmark Scale factor will re-adjust accordingly.
5. Finish specifying options for your test, or click **Save/Close** to save your modifications and close the wizard.

Benchmark Scale Factor and User Load Relationship

The TPC-C and the Scalable Hardware standard benchmark tests have a scale factor / user load relationship.

- For the TPC-C benchmark, the ratio is 10 virtual users per scale factor.
- For the Scalable Hardware benchmark, the scale factor is also the number of users that will return data from their queries.

If you attempt to create a benchmark test using a scale factor / user load ratio outside of these specifications, Benchmark Factory warns you.

ℹ **Note:** You can set a default Benchmark Scale factor to use for each standard benchmark test. Go to **Edit | Settings | Benchmarks | *benchmark type***.

Options Tab (Create Objects Step)

Use the Options tab to specify how to create benchmark objects and load data for the Create Objects step of a standard benchmark test. You can also specify how to retain/maintain the tables after the test is finished.

You can specify a default setting for Object/Data Retention in **Edit | Settings | Benchmarks**. See Benchmarks Settings - General on page 152 for more information.

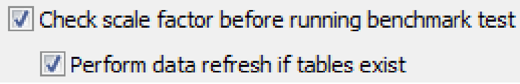
To specify options for loading benchmark tables

1. In the Edit Job or New Job Wizard, under **Workload** select the **Create Objects for *benchmark*** step for the selected benchmark test.
2. From the Test Options tab, select the Options tab.
3. Use the Options tab to specify options for creating objects and loading data. Review the following for additional information:

Distribute index creation (one index create per virtual user)	**Select** to use multiple virtual users to create indexes at the same time (one virtual user for each index). ℹ **Tip:** For IBM DB2 databases, you might encounter deadlock errors if database configuration is incorrect. If you do not select this option, all indexes are created by a single virtual user. ℹ **Note:** To specify parallel index creation for indexes, you can customize the SQL used to create indexes. See Customize SQL for Creating Objects on page 96 for more information.
Distribute Load using Benchmark Factory Agents	You can choose to use console or agents to load benchmark data. • **Select** to distribute data generation across multiple agent machines to speed up the standard benchmark data load. • If you do not select this option, the console is used to load standard benchmark data. ℹ **Important Note:** Using the console for large data loads could slow down database load times significantly. Use only for small scale factors.
Number of virtual users to perform creation/load	Specify the number of virtual users to use to load table data. The number of virtual users used per table will depend on the table size. ℹ **Notes:** • Benchmark Factory recommends using a number

	that is a multiple of the number of tables in the selected standard benchmark test. Refer to benchmark test specifications for the table count. • You cannot specify a number less than the number of tables in the selected standard benchmark test.

4. Select an **Object/Data Retention** method. Review the following for additional information:

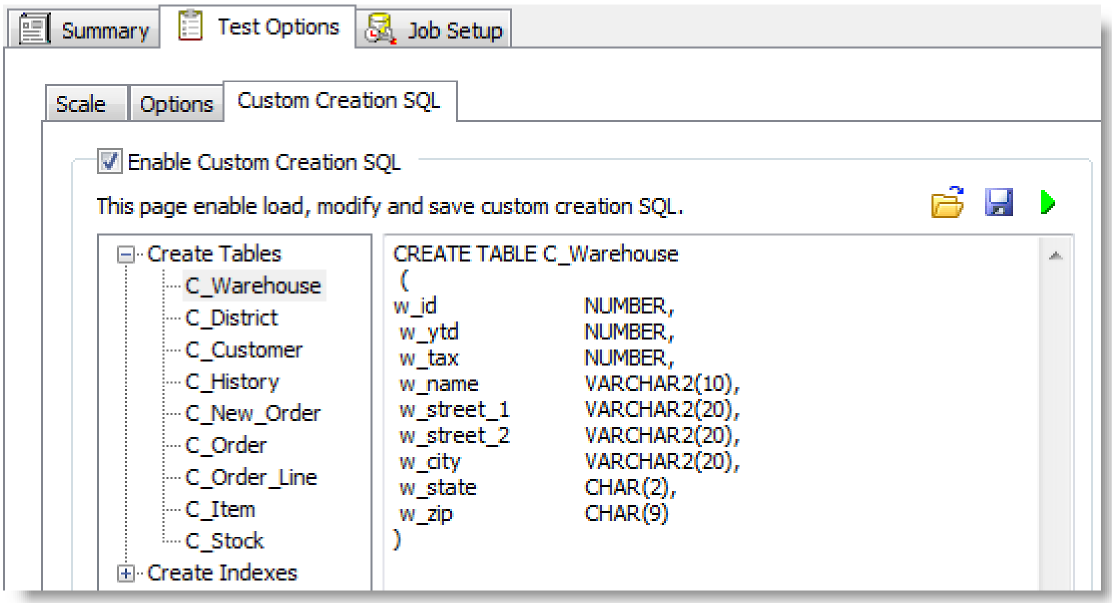
Create/Load objects if objects don't exist (no backup sets created)	If objects do not exist, Benchmark Factory creates the objects and loads data. If you select this option, Benchmark Factory does not create backup tables.
Delete benchmark objects after completion of job	Benchmark Factory deletes objects after a job is finished.
Refresh data using backup sets	When backup sets exist, Benchmark Factory reloads data and recreates indexes. If backup sets do not exist, Benchmark Factory creates objects and loads data. Then Benchmark Factory creates the backup tables.
Recreate objects and reload data every execution	Benchmark Factory always deletes the existing objects and then recreates/reloads for each job execution. This allows you to ensure that the dataset is always in initialized status.
Refresh data using inserts	If data does not exist, Benchmark Factory uses bulk insert to load data. If the following Benchmark Settings are selected,  ☑ Check scale factor before running benchmark test ☑ Perform data refresh if tables exist and the Benchmark Scale factor of the existing tables does not match the new requirement, the process of inserting additional data or deleting data my run long, depending on the size of the existing data and number of indexes. **Note:** If database does not support bulk insert, insert is used.

Customize SQL for Creating Objects

Use the Custom Creation SQL tab to customize the SQL used to create objects in an Industry Standard Benchmark test. You can save your modified SQL as a .xml file to reuse the same SQL and settings in multiple tests.

To use custom SQL to create objects

1. In the New Job Wizard, after adding an industry standard benchmark test, select the Create Objects step.
2. Select the Test Options tab and then open the Custom Creation SQL tab.
3. Select the **Enable Custom Creation SQL** option.



☑ Enable Custom Creation SQL

This page enable load, modify and save custom creation SQL.

- Create Tables
  - C_Warehouse
  - C_District
  - C_Customer
  - C_History
  - C_New_Order
  - C_Order
  - C_Order_Line
  - C_Item
  - C_Stock
- Create Indexes

```
CREATE TABLE C_Warehouse
 (
w_id                 NUMBER,
 w_ytd               NUMBER,
 w_tax               NUMBER,
 w_name              VARCHAR2(10),
 w_street_1          VARCHAR2(20),
 w_street_2          VARCHAR2(20),
 w_city              VARCHAR2(20),
 w_state             CHAR(2),
 w_zip               CHAR(9)
)
```

4. To modify the SQL, select an object in the left pane, and then edit the SQL in the right pane.

   - To test run the modified SQL, click ▶.
   - To restore the default SQL for the selected object, click **Restore SQL**.
   - To restore the default SQL for all objects, click **Restore All**.

   Your edits are saved automatically if you click **Save/Close** or run the job.

5. To save the modified SQL as an .xml file, click 💾. Specify a file name and click **Save**. The SQL for the objects you modified is saved to the file.

   Note: By default, the .xml file is saved to the Data directory. To change the location of the Data directory, go to **Edit | Settings | General**.

6. To import custom SQL from a previously-saved .xml file, click 📂.

   Note: When importing SQL from a file, Benchmark Factory uses the default SQL for all objects not in the .xml file.

7. To save your changes, click **Save/Close** or specify the remaining wizard options and run the job.

8. To retain but disable the custom SQL in the Custom Creation SQL tab, clear the **Enable Custom Creation SQL** checkbox.

Note: If you modify an object name, column name, or a data type, this change could prevent data/objects from loading successfully or could cause the standard benchmark test to fail.

Replication Table Options Tab

Use the Table Options tab of the Replication benchmark test to design the structure of tables used in this test. You can specify the number of tables, the number of columns per table, column data types, and the proportion of each data type per table.

You can specify default values for these Replication table options at **Edit | Settings | Benchmarks | Replication**. The default values are applied when you create a new Replication test. See Replication Benchmark Settings on page 155 for more information.

To specify Replication table options

1. In the New/Edit Job wizard, select the **Create Objects for Replication** step under the Workload section.
2. Select the **Test Options** tab and then the **Table Options** tab.
3. To define the Replication tables, specify the following options.

Number of Tables	Select the number of tables to include in this Create Objects for Replication step.
Number of Columns per Table	Specify a range. The number of columns in each table will be randomly selected (using Uniform distribution) from this range.

4. Then use the grid to specify the data types to include in the tables, as well as the proportion of each data type. The grid initially displays the data type list and default values that are specified in the Replication Table Structure page of the Settings dialog (**Edit | Settings**).

   To define a new data type for this test, click **Add**. Then specify data type details by entering values in the grid. Review the following for additional information:

DataType column	Select a data type from the list (click inside the column to display the drop-down list).
Weight column	Specify a weight for this data type. This value is used by Benchmark Factory to determine what percentage of the columns will be defined with this type. The weight/percentage is applied to each table (if possible) and to the database as a whole. The total weight is displayed below the grid.

5. Click **Add** to add another data type. Click **Delete** to delete the selected data type. Continue this process until you have defined all the data types (columns) you want to include in the Replication tables.

ℹ **Notes:**

- Replication tables will automatically include a primary key column.
- The number of columns per table is determined using a Uniform distribution model.
- Right-click a data type in the grid to perform a copy and paste action.

Test Options - Transactions

Transactions Tab

The Transactions tab of the Edit Job or New Job Wizard allows you to modify the transaction mix, add or delete user scenarios, add or delete transactions, or modify SQL for the selected test in the job.

To modify the transaction mix in a test

1. In the Edit Job or New Job Wizard, select a test under **Workload** in the left pane.
2. From the Test Options tab, select the Transactions tab.
3. Use the Transactions tab to modify transactions and/or transaction mix for the selected test. Review the following for additional information:

✏ **Edit**	Click to edit the selected user scenario or transaction. The

	scenario/transaction opens in a dialog for editing. When editing a scenario or transaction, you can: • Review the list of SQL in the user scenario. • Review the benchmark transaction. • Add additional transactions to the user scenario. • Delete SQL or transactions from the user scenario. • Edit a SQL statement or its bind parameter. • Specify latency for each SQL, each transaction, or for the user scenario. See About Latency on page 143 for more information.
⬆ ⬇	Move the selected item up or down in the list in the Transaction tab.
Transaction Tab Columns	Review the following column descriptions for the Transaction tab. **Weight column**—(Transactions) The likelihood a transaction executes by a virtual user. (Not available in Test SQL for Scalability.) • **Transaction weight definition:** If a test has two transactions, A and B, with A having the weight of one and B having the weight of six, transaction B will run six times for every time A runs once. The run order is randomly generated for users. **Percentages column**—(Transactions) Transaction's weight as a percentage or the total weight. (Not available in Test SQL for Scalability.) **Users**—(User scenarios) Displays the number of users tested during replay for this scenario. Double-click to edit. **Executions**—(User scenarios) Displays the number of executions for this scenario. Double-click to edit.
Ave. number of statements per commit	(Replication test only) Specify the number of statements to include in a commit. Use this option to attempt to simulate typical transactions.
Add SQL Transactions	Add additional transactions to the test. Select one of the following from the drop-down list and then click ⊕. • ***Add SQL Transactions***—Add a SQL statement. You can enter SQL or import SQL from a file. See Add SQL Transactions on page 104 for more information. ◦ Or click 📝 to launch the BF Script Wizard. See BFScript Wizard on page 198 for more information. • ***Add Stored Procedures***—Import SQL from a stored procedure. • ***Import Benchmark Transactions***—Import one or more transactions from one of the standard benchmarks.
Add User Scenario	Add a new user scenario to the test. Select one of the following from the drop-down list and then click ⊕. • ***Add User Scenario***—Add a new user scenario and then populate it with transactions (from one of the categories listed above). ℹ **Note:** You can right-click a User Scenario and select **Replace Child Latencies**. See Replace Child Latencies on page 139 for more information. • ***Add From Session Metafile***—Add a user scenario by importing a transaction file.

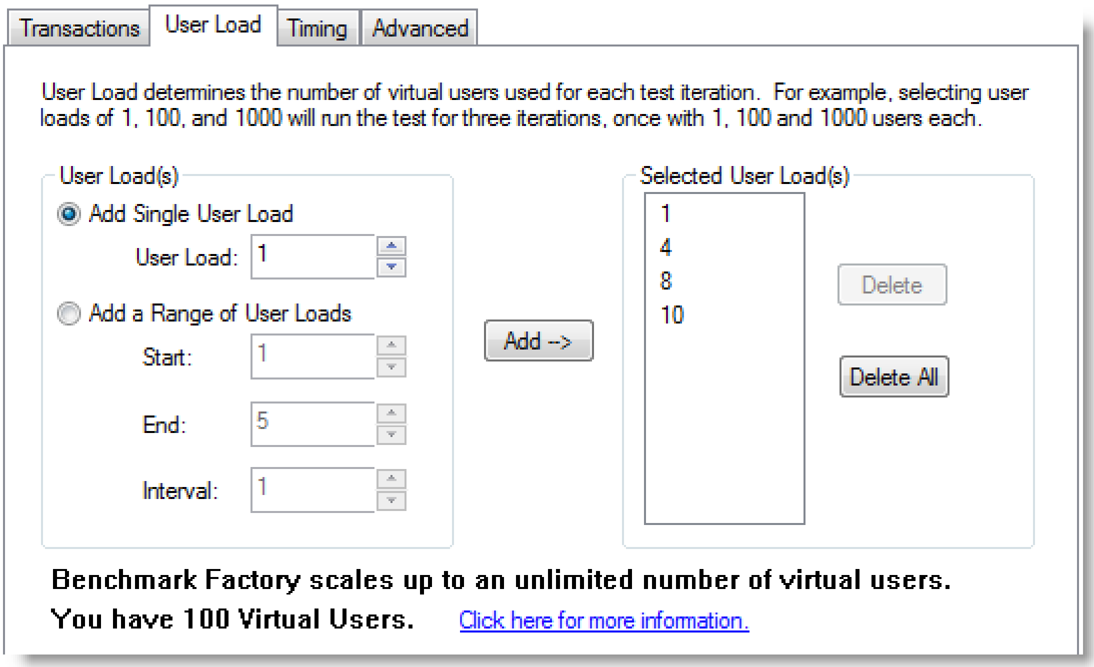
Replay User Load	Add a new scenario to the test by importing a workload from an Oracle trace file, a SQL Server trace table, an ODBC trace file, or a text file.

Note: New user scenarios/transactions are added to the top of the list. Use the up and down arrows to rearrange the order of items in the list.

Specify User Load

The User Load tab of the New Job or Edit Job Wizard allows you to specify the number of virtual users for each iteration of a test.

Note: To set default values for User Load, go to **Edit | Settings | User Load**. If you modify the default settings, the changes will apply to new jobs only, not to existing jobs.



User Load determines the number of virtual users used for each test iteration. For example, selecting user loads of 1, 100, and 1000 will run the test for three iterations, once with 1, 100 and 1000 users each.

User Load(s)

● Add Single User Load

User Load: 1

○ Add a Range of User Loads

Start: 1

End: 5

Interval: 1

Selected User Load(s)

1
4
8
10

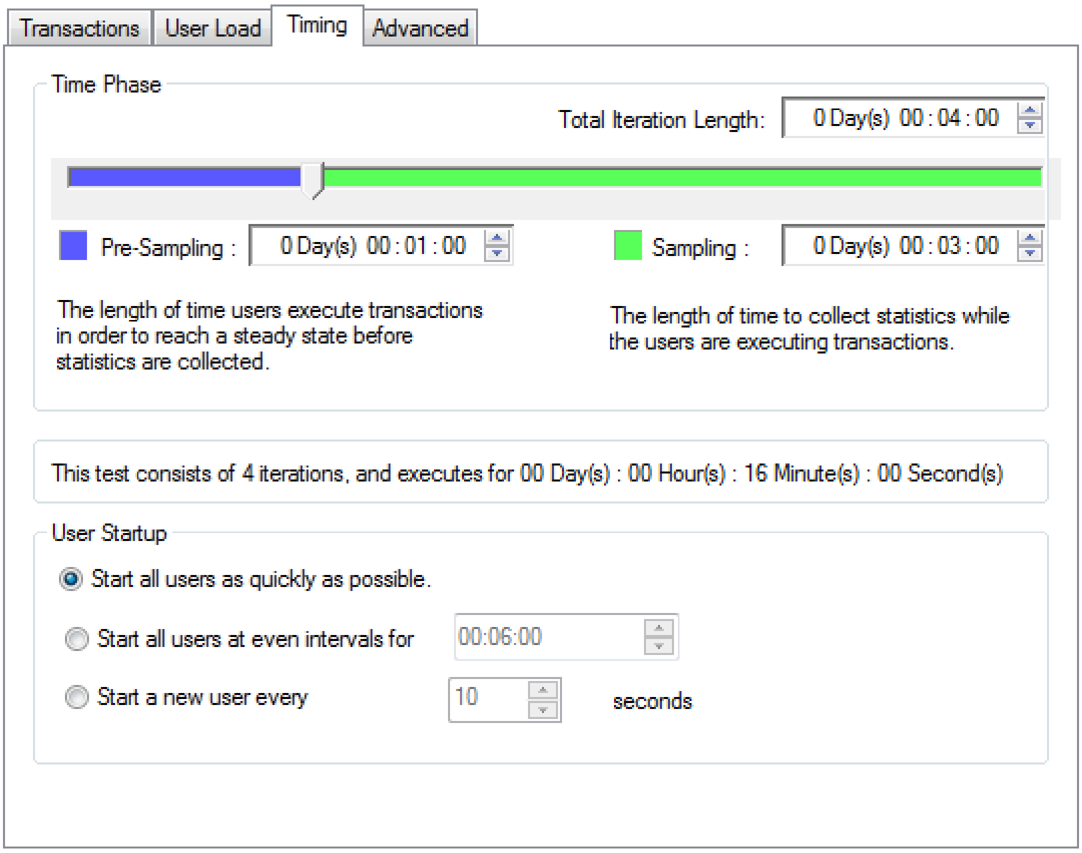
Benchmark Factory scales up to an unlimited number of virtual users.

You have 100 Virtual Users. Click here for more information.

Section	Description
User Load	**Add Single User Load**—Specify a single user load, then click **Add** to add it to the selected user loads. **Add a Range of User Loads**—Specify a range, then click **Add** to add the range to the selected user loads.

Timing Tab

Use the Timing tab to specify timing phases associated with a test.



Time Phase

Total Iteration Length: 0 Day(s) 00 : 04 : 00

🟦 Pre-Sampling : 0 Day(s) 00 : 01 : 00

🟩 Sampling : 0 Day(s) 00 : 03 : 00

The length of time users execute transactions in order to reach a steady state before statistics are collected.

The length of time to collect statistics while the users are executing transactions.

This test consists of 4 iterations, and executes for 00 Day(s) : 00 Hour(s) : 16 Minute(s) : 00 Second(s)

User Startup

● Start all users as quickly as possible.

○ Start all users at even intervals for 00:06:00

○ Start a new user every 10 seconds

Section	Field/Icon	Description
Time Phase	Total Iteration Length	The composite of the Pre-Sampling and Sampling time.
	Pre-Sampling	The length of time users execute transactions in order to reach a steady state before statistics are collected.
	Sampling	The length of time used to collect statistics while the users are executing transactions.
User Startup	Start all users as quickly as possible	Starts all users immediately after a test begins.
	Start all users at even intervals for	Sets the amount of time in which to start all users at even intervals. The interval duration is equal to this value divided by the number of users.
	Start a new user every [n] seconds	Starts a new user, then waits the entered number of seconds before starting the next user.

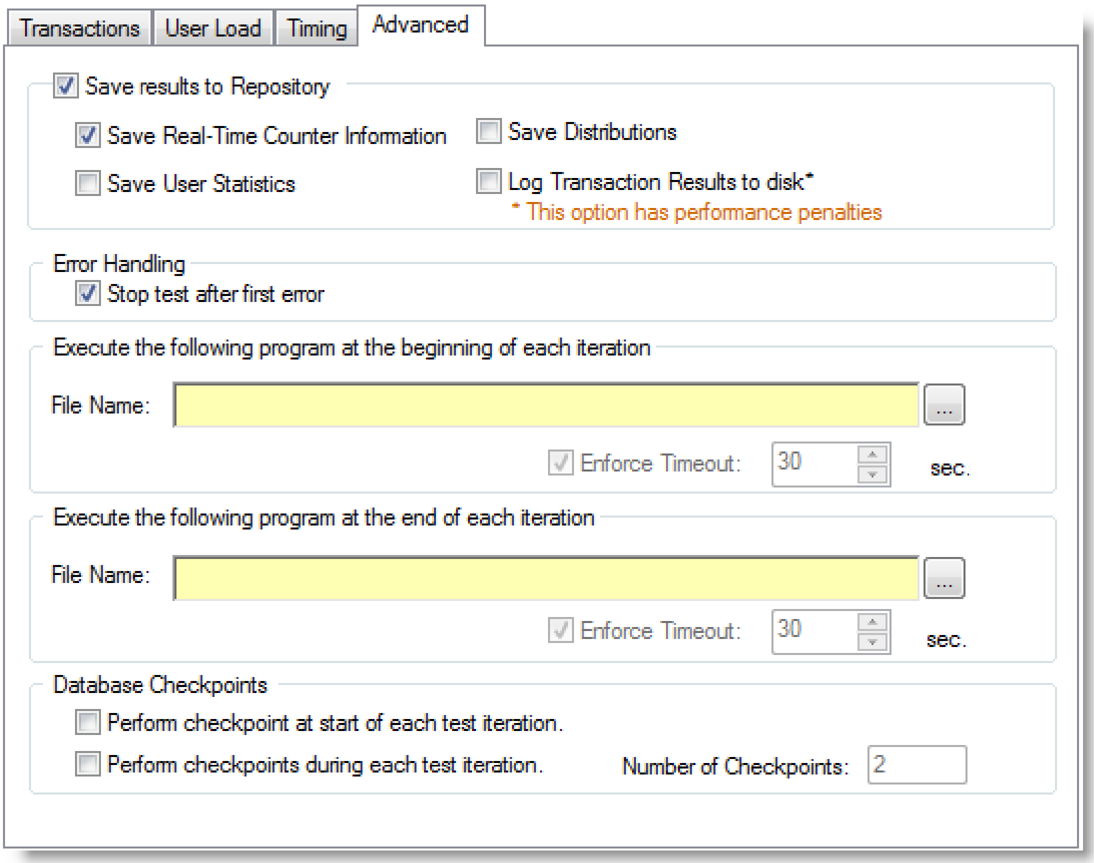
Advanced Tab

You can use the Advanced tab to specify the information to save in the Repository during a test run, to set error handling properties, and to specify database checkpoints for the test.

The repository is a database where all Benchmark Factory information is saved. It contains information about the transactions stored in each benchmark DLL and statistical information from benchmark executions.

To specify Advanced options

1. In the New/Edit Job Wizard, select a test under **Workload**.
2. Select the Test Options tab. Then select the Advanced tab.



☑ Save results to Repository

☑ Save Real-Time Counter Information ☐ Save Distributions

☐ Save User Statistics ☐ Log Transaction Results to disk*

* This option has performance penalties

Error Handling

☑ Stop test after first error

Execute the following program at the beginning of each iteration

File Name:

☑ Enforce Timeout: 30 sec.

Execute the following program at the end of each iteration

File Name:

☑ Enforce Timeout: 30 sec.

Database Checkpoints

☐ Perform checkpoint at start of each test iteration.

☐ Perform checkpoints during each test iteration. Number of Checkpoints: 2

3. Specify advanced options. Review the following for additional information:

Section	Description
Save results to Repository	Select to save test information to the Repository. Then select one or more of the following options: • **Save Real-Time Counter Information**—Saves real-time performance monitoring information to the Repository during the test. See Performance Counters Tab on page 111 for more information. **Note:** To specify sampling rate, go to **Edit | Settings | Statistics | Real-Time counters**. • **Save User Statistics**—Saves user-level statistics. • **Save Distribution**—Saves transactions per second and response time per user distribution. • **Log Transaction Results to disk**—Logs the transaction results of a job to disk.

Section	Description
	Note: Checking this option imposes performance penalties.
Error Handling	**Stop test after first error**—The test is stopped when an error is reported. You can specify a default setting in **Edit | Settings | Error Handling**. See Error Handling Settings.
Execute the following program at the beginning of each iteration	(Available only for benchmark tests and some capture/replay tests.) Browse to and select the file to execute at the beginning of each iteration. **Note**: This field accepts BFScripts. **Enforce Timeout**—Select and enter a time to enforce a timeout on the file executing.
Execute the following program at the end of each iteration	(Available only for benchmark tests and some capture/replay tests.) Browse to and select the file to execute at the end of each iteration. **Note**: This field accepts BFScripts. **Enforce Timeout**—Select and enter a time to enforce a timeout on the file executing. If the file does not complete in the specified time, it is stopped and the job continues.
Database Checkpoints	**Perform checkpoint at start of each test iteration**—Initiates a database checkpoint at the beginning of a test iteration. **Perform checkpoints during each iteration**—Initiates a database checkpoint during a test iteration. **Number of checkpoints**—Specifies the number of checkpoints to initiate.

Options Tab (Capture/Replay)

Settings on the Options tab allow you to apply scaling to the user scenarios (transactions) for a particular test.

To apply scaling to user scenarios

1. In the Edit Job or New Job Wizard, select a test under **Workload** in the left pane.
2. From the Test Options tab, select the Options tab.
3. Modify options. Review the following for additional information:

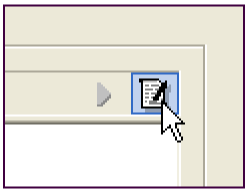
Enable Userload scaling (simulation)	Select to enable scaling for all user scenarios in the selected test. • Move the slider to the right to scale up userload for all scenarios (transactions) during playback. • You can also specify the number of users per scenario in the Transactions tab. See Transactions Tab on page 98 for more information.
Execute by time	Sets the time and length of the capture replay test.

Add SQL Transactions

The *Add SQL Transactions* function allows you to add a SQL statement transaction to a test in the Job workload.

To add a SQL statement transaction to a test

1. In the Edit Job or New Job Wizard, select a test under **Workload** in the left pane.
2. From the Test Options tab, select the Transactions tab.
3. Select *Add SQL Transactions* from the **Add SQL Transactions** drop-down list.
   - Or select a user scenario and click ✎ **Edit** to open the list of SQL transactions. Then select *Add SQL Transactions*.
4. The Add SQL Transaction dialog opens. Do one of the following:
   - Enter a SQL Statement.
   - Or click 📝 to launch the BF Script Wizard. See BFScript Wizard on page 198 for more information.



   - Or click 📂 to import SQL from a file.
5. Modify the remaining options on the SQL Statement tab. Review the following for additional information:

Statement Name	Enter a name for the statement, or use the default.
Execution Method	Select an execution method for the SQL statement. • Prepare and Execute SQL—Prepare the SQL and execute immediately after preparation. • Prepare SQL Only—Prepare SQL without executing. • Execute already prepared SQL—These transactions will run the SQL statement contained with the reference prepared SQL when called. If the statement has not been prepared, an error will generate.
▶	Click to execute the SQL statement. See Run SQL Preview on page 139 for more information.
▦	Click to add a bind variable. • Select the Bind Parameters tab and then click ▦ or double-click within the Bind Parameters window to add a bind variable and value.

6. Select the Latency tab to specify latency values for the SQL transaction. See About Latency on page 143 for more information.
7. Click **OK** in the Add SQL Transaction dialog to the save the SQL statement and add it to the test.

BFScript Wizard

ℹ **Note**: Controls with yellow background can be scripted.

Benchmark Factory scripting allows you to randomize the load testing process. Scripting can be utilized by selecting the SQL Statement node and inserting the script in the properties view. A field has scripting capabilities if its right-click menu contains the Script Wizard option.

The Script Wizard is a quick and easy way to use Benchmark Factory scripts. The Script Wizard consists of a script category and a script function. Each script function has a short description included and if applicable, its parameters.

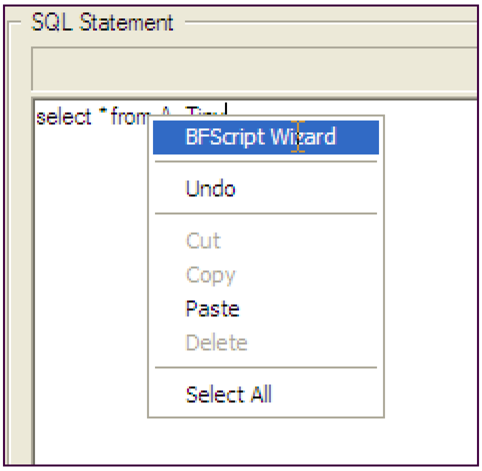
There are two features within Benchmark Factory that assist you when using scripting capabilities:

- Script Wizard
- Script Assist

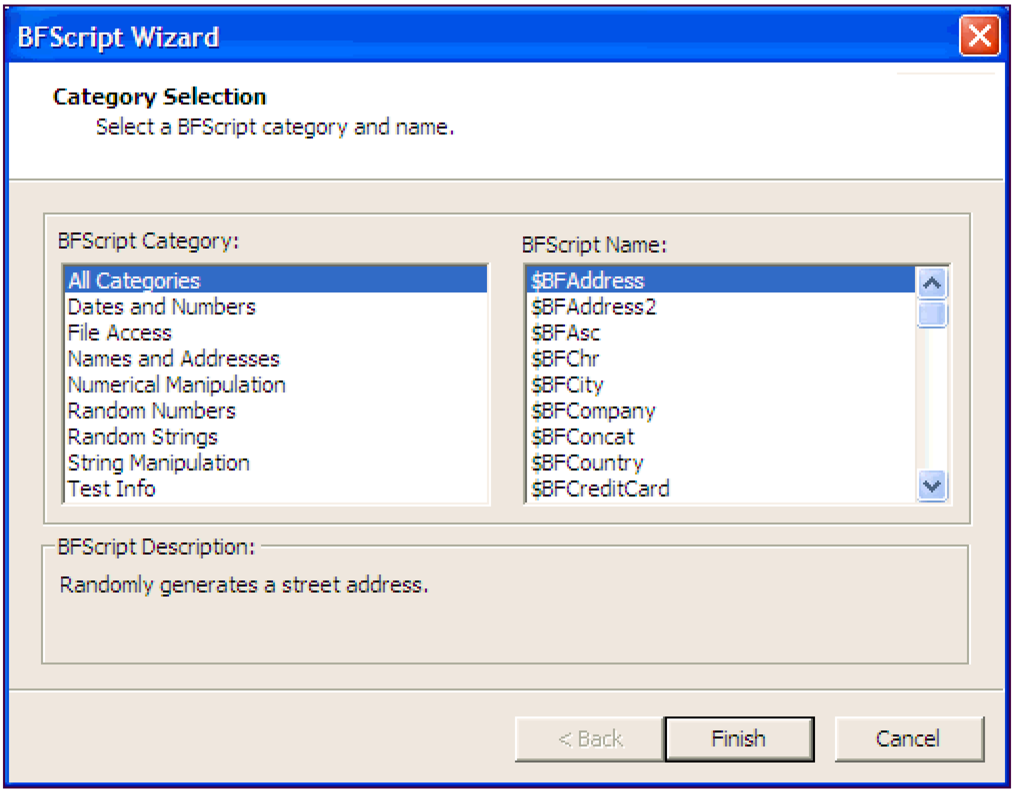
Click here for an overview of $BFScripts.

Script Wizard

1. Right-click the edit control. A drop-down display.



2. Select **BFScript Wizard** from the drop-down. The Script Wizard displays.



BFScript Wizard

Category Selection

Select a BFScript category and name.

BFScript Category: All Categories

BFScript Name: $BFAddress

BFScript Description: Randomly generates a street address.

3. Select the desired script.
4. Some scripts require further parameters settings. For a complete overview of individual script settings, please refer to Scripts Overview.

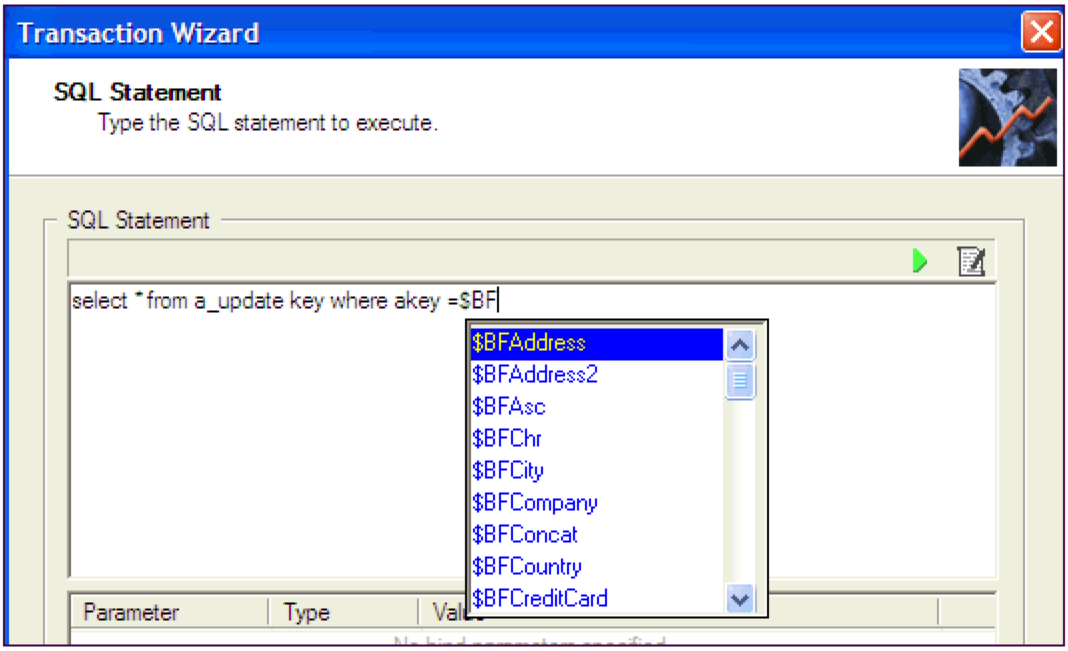
   If scripts require parameters:

1. Click **Next.**
2. Enter the required parameters.

5. Click **Finish**. The script and parameters display in the Script Wizard.

Script Assist

1. Type in the desired SQL statement. When $BF is typed the Script Assist is activated allowing you to insert the desired script.



```
select * from a_update key where akey =$BF
```

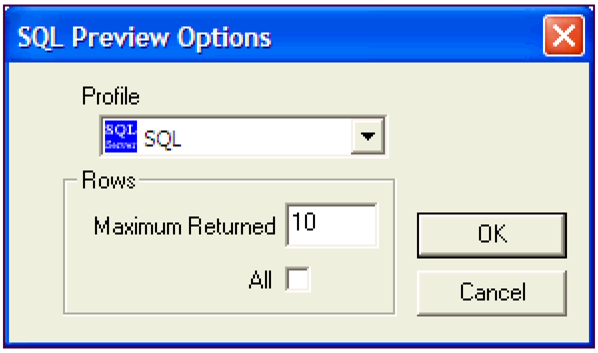
2. Select the desired script.
3. Press **Enter**. The script displays.
4. Type in parameters as required.

Run SQL Preview

SQL Preview allows you to preview the result set of an individual SQL statement when adding or editing a SQL transaction.

To run SQL Preview

1. In the SQL Statement tab, click ▶.
2. Select a connection from the drop-down list.



Profile: SQL

Maximum Returned: 10

All □

4. In the **Maximum Returned** field, enter the maximum number of rows to preview, or select the **All** option to preview all rows.
5. Click **OK**. The SQL Preview window opens.
6. Review the information.
7. Click **Close**.

About Latency

Latency is delay added to the execution of a transaction to control of how fast transactions are submitted to the system-under-test. It is used to either make the transaction execution rate more like real-world executions or to control the transaction rate. This delay can be added to the beginning and/or end of a transaction execution.

You can modify latency values for a transaction by editing the transaction in the Transactions tab of the New/Edit Job wizard. In addition, you can specify latency values at the time you add transactions/scenarios to a test using the Transactions tab.

Benchmark Factory allows you to set default latency values for the transactions you add.

To modify latency values

1. Open the New Job wizard for a new job or the Edit Job wizard for an existing job.
2. Select the test (or the step) under **Workload** that contains the transactions.
3. Select the **Test Options** tab. Then select the **Transactions** tab.
4. Select the transaction or scenario for which you want modify latency and click **Edit**.
5. In the dialog that opens, select the **Latency** tab. For a scenario, to modify latency for individual transactions, select a transaction and click **Edit**, and then select the Latency tab.
6. Modify latency options. To specify no latency, select **No Delay**. To add latency, select one of the other options. Review the following for additional information:

No Delay	Select to specify no latency.
Transaction Rate	Select to set a transaction rate by specifying the interval between successive transactions arriving at the server. **Interarrival Time**—Select one of the following: • Absolute—Select to specify a fixed interarrival time. Then enter a value. • Distribution models—To use a distribution model for determining interarrival time, select a model and enter the requested value.
Real-World Delays	Use this option to simulate real-world processing by inserting a delay before and/or after each transaction execution. For each delay type, you can insert a fixed time by selecting *Absolute*, or you can select one of the distribution models for specifying time. **Keying Time**—Inserts a delay before execution to simulate the amount of time spent performing data entry. **Think Time**—Inserts a delay after execution to simulate the amount of time "to think" about the results of the previous transaction.

7. Click **OK** to save your changes and close the dialog.
8. Then click **Save/Close** in the wizard to save your changes.

Why Use Latency?

Real-World Delays: Add Keying Time and Think Time to create a workload that accurately simulates real-world users, as opposed to a workload that submits transactions as fast as possible. Using real-world delays allows for easier interpretation of the results. If you do not add delay, it might be difficult to equate *N virtual users running with no delay* to an accurate number of real users.

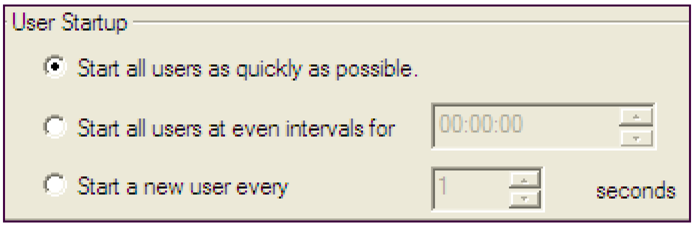
Interarrival Time: Specify Interarrival Time to create a workload in which transactions arrive at the system-under-test at a specific interval. This method allows you to set the timing relationship between executing transactions and is commonly used when replaying a captured workload.

Stagger User Startup

When running a load test, you have the option of staggering user startup. This allows you to adjust startup loads if, for example, you are having issues with overloading of servers when all of the users attempt to start at the same time.

To stagger user startup

1. In the Edit/New Job Wizard, select a test under **Workload**.
2. Select the **Timing** tab (under Test Options tab).
3. In the User Startup options, select the desired parameters.



User Startup

● Start all users as quickly as possible.

○ Start all users at even intervals for 00:00:00

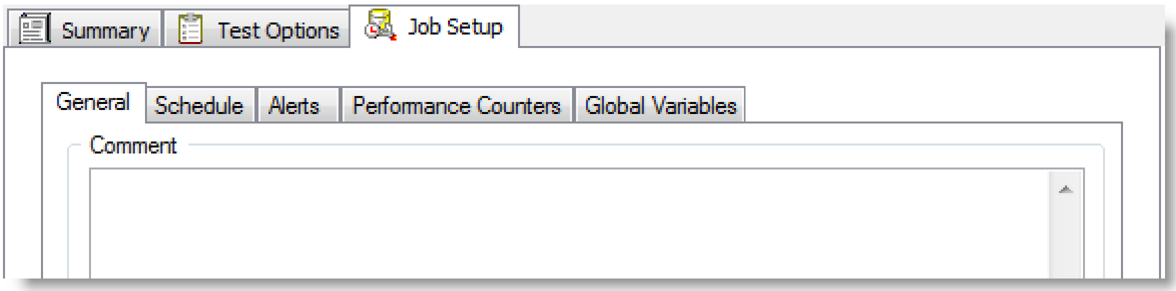
○ Start a new user every 1 seconds

- **Start all users as quickly as possible**: Starts all users immediately after a test begins.
- **Start all users at even intervals for**: Sets the amount of time in which to start all users at even intervals. The interval duration is equal to this value divided by the number of users.
- **Start a new user every**: Starts a new user, then waits the entered number of seconds before starting the next user.

Job Setup Options

Job Setup Tab

Use the Job Setup tab of the New Job wizard or Edit Job wizard to schedule a job, set up email notifications (alerts), add performance counters, and manage global variables.



To specify Job Setup options

1. In the New/Edit Job wizard, after configuring the workload, select the **Job Setup** tab.
2. Use this tab to specify job-level options, such as the job schedule, alerts, or performance counters.

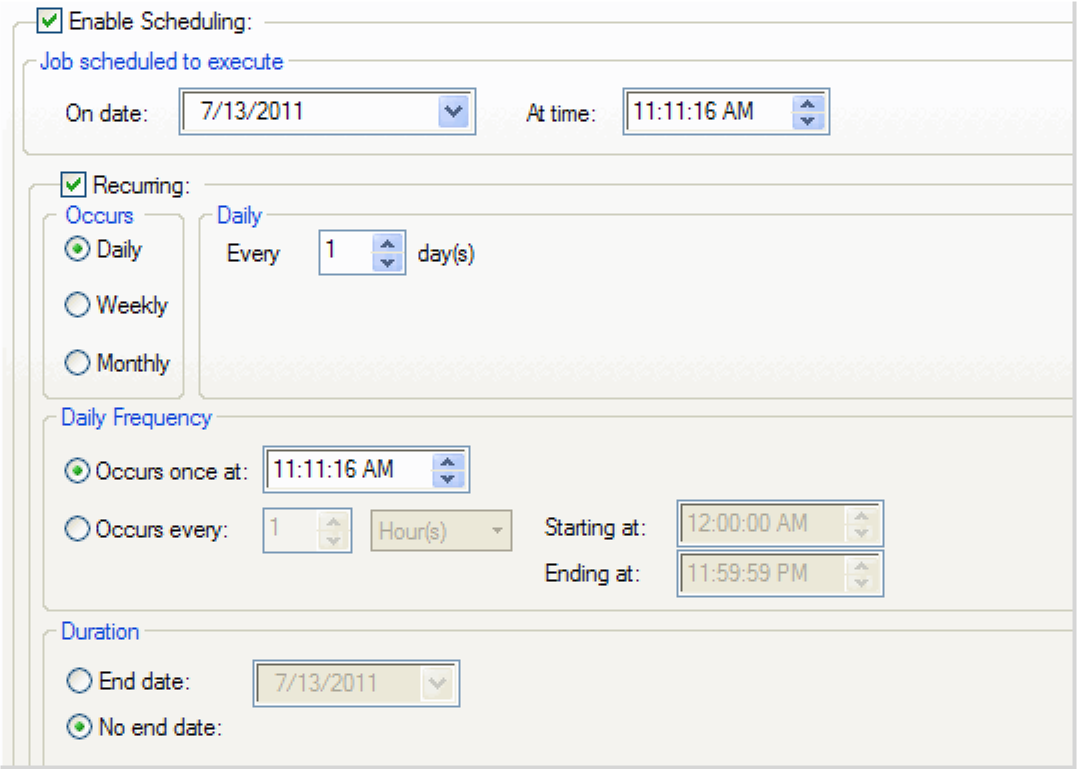
Review the following for more information:

General tab	Enter a comment or specify error-handling behavior.
Schedule tab	See Schedule a Job.
Alerts tab	See Alerts - Job Setup.
Performance Counters tab	See Performance Counters Tab.
Global Variables tab	See Global Variables Tab.

Schedule a Job

Benchmark Factory allows you to schedule a job to run at a future time. Scheduling allows you to set predetermined dates, recurrences, daily frequencies, and durations of jobs.

Note: When using the recurring option, use Ready Job in place of Run Job.



☑ Enable Scheduling:

Job scheduled to execute

On date: 7/13/2011 At time: 11:11:16 AM

☑ Recurring:

Occurs: ● Daily ○ Weekly ○ Monthly

Daily: Every 1 day(s)

Daily Frequency

● Occurs once at: 11:11:16 AM

○ Occurs every: 1 Hour(s) Starting at: 12:00:00 AM Ending at: 11:59:59 PM

Duration

○ End date: 7/13/2011

● No end date:

To schedule a job

1. In the New/Edit Job Wizard, select a test under **Workload**.
2. Select the **Job Setup** tab.
3. Select the **Schedule** tab.
4. Select the **Enable Scheduling** checkbox.
5. To run the job once, select a date and time to execute the job.

6. To run the job on a recurring basis, select a start date and time, and select the **Recurring** check box.
    a. Then specify a frequency.
    b. To run the job multiple times per daily, specify options under Daily Frequency.
7. Select an end date, or select the **No end date** checkbox.
8. Click **Save/Close** to save your options and close the wizard.

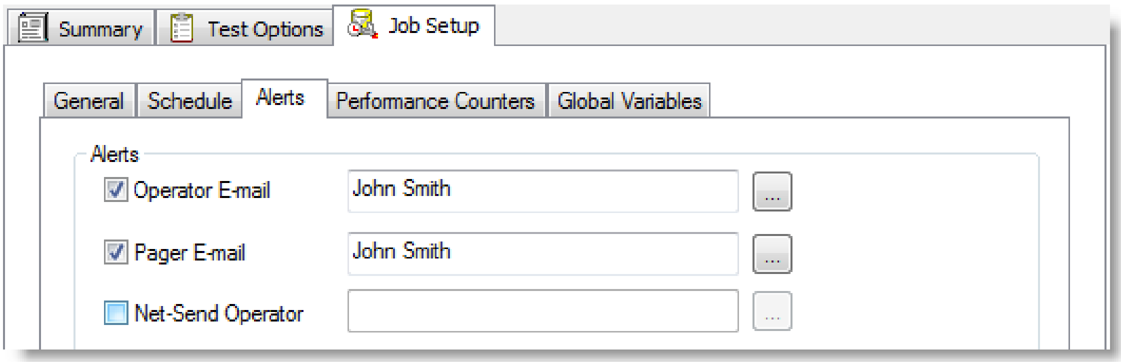
Alerts - Job Setup

Use the Alerts tab to set up automatic notification of job completion.

Note: Some email programs may prompt for send message confirmation. The job will not complete until confirmation is received.

To setup job notification

1. In the New/Edit Job Wizard, select a test under **Workload**.
2. Select the Job Setup tab.
3. Then select the Alerts tab.



☑ Operator E-mail: John Smith ...
☑ Pager E-mail: John Smith ...
□ Net-Send Operator: ...

4. Select one or more notification types. Review the following for additional information:

Operator E-mail	Sends an email containing a file attachment summarizing the results of a job.
Pager E-mail	Sends notification of job completion.
Net-Send Operator	Sends a network message to a specific machine notifying that a job is complete.

Performance Counters Tab

Performance counters keep track of statistics during test execution. By default the following performance counters are recorded when a job executes, even though they do not display in the Performance Counter tab for a connection or job.

- AVG_TIME
- BPS
- DEADLOCKS
- TOTAL_ERRORS
- MAX_TIME

- RPS
- TOTAL_BYTES
- TOTAL_ROWS
- TPS
- USERLOAD

You can add performance counters to a connection or a job.

To add performance counters to a connection

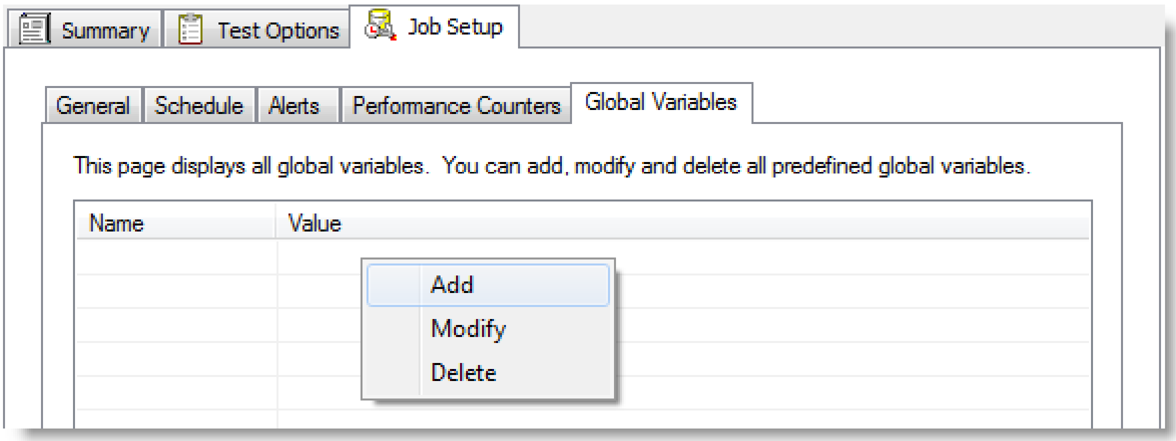
1. In the New Connection dialog when creating a new connection (or the Connection dialog for an existing connection), select the Performance Counters tab.
2. To add a new counter, click 📊.
3. Select a computer from the drop-down list.
4. To display counters, expand an object node by clicking the node's down-arrow.
5. Select one or more counters and click **Add**.
6. When you have finished adding counters, click **Close** to close the Browse Performance Counters dialog.
7. To delete a counter, select the counter and click ✕.

Global Variables Tab

From the global variable tab, you can add or delete the required variables.

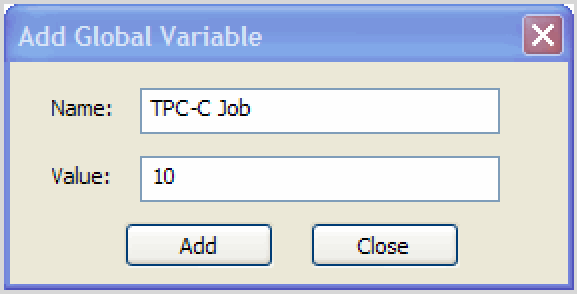
To add a global variable

1. Right-click to display the global variable dialog.



This page displays all global variables. You can add, modify and delete all predefined global variables.

2. Enter the name and value.

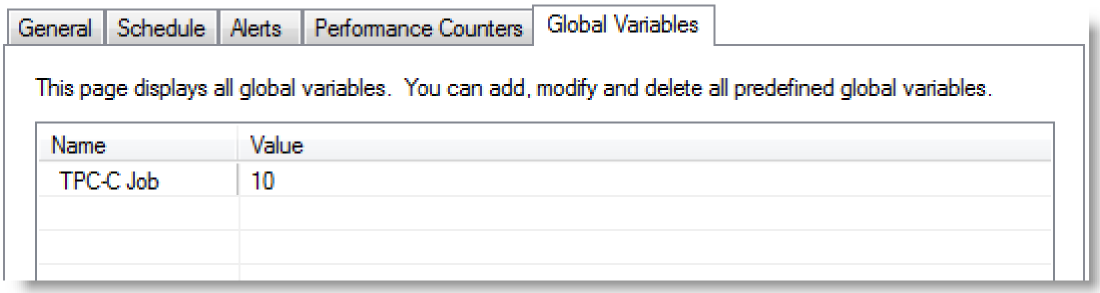


Name: TPC-C Job

Value: 10

3. Click **Add**.

   The new global variable displays.



This page displays all global variables. You can add, modify and delete all predefined global variables.

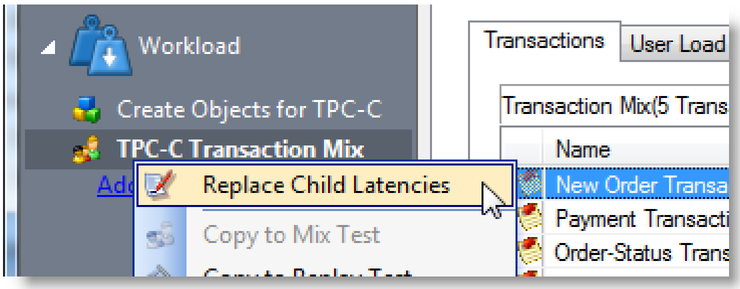
Name	Value
TPC-C Job	10

Replace Child Latencies

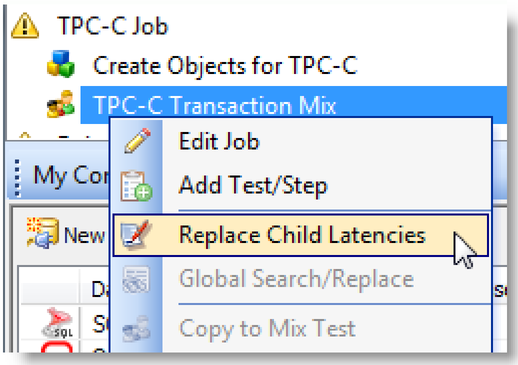
Latency is delay added to the execution of a transaction to control of how fast transactions are submitted to the system-under-test. You can use the **Replace child latencies** dialog to quickly modify latency for all the direct children of a test. For example, when an Oracle trace file is imported, the latencies are automatically set to replay the transactions at their original rate, but changing these to a set value can replay the transactions in a shorter period of time.

To replace child latencies at the test level

1. Use one to the following methods to open the Replace Child Latencies dialog:
   - In the New/Edit Job Wizard, right-click a test and select **Replace Child Latencies**.

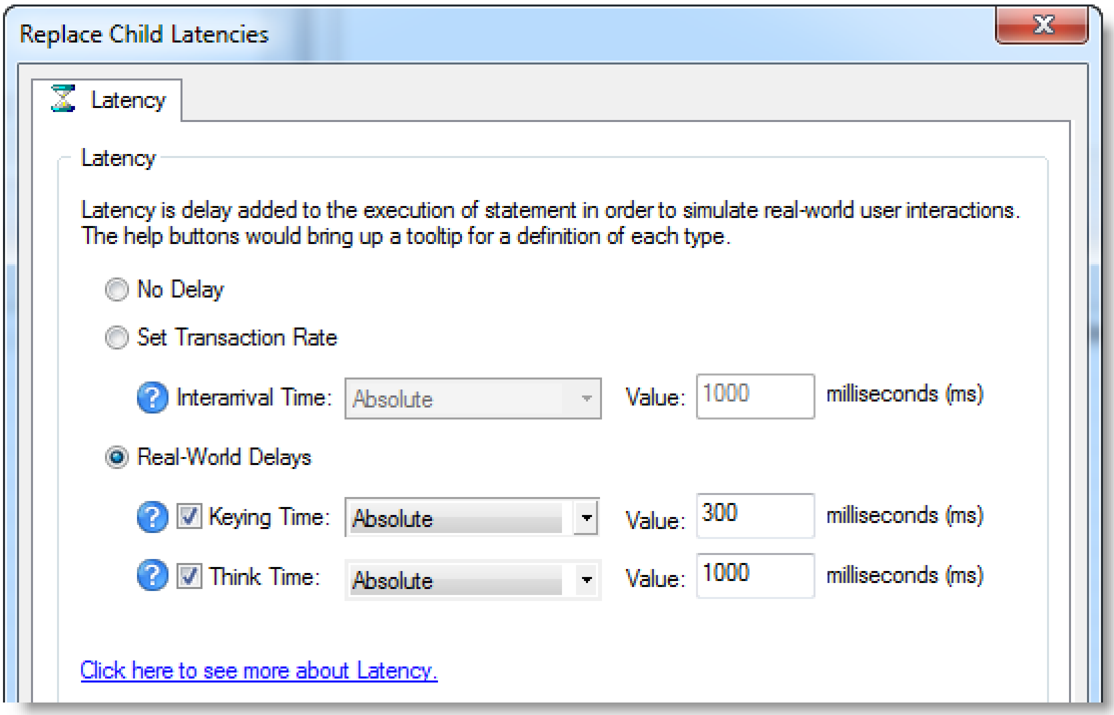


   - In the Jobs View pane, right-click a test and select **Replace Child Latencies**.



2. In the Replace Child Latencies dialog, modify latency options. See About Latency on page 143 for more

information.



Latency is delay added to the execution of statement in order to simulate real-world user interactions. The help buttons would bring up a tooltip for a definition of each type.

○ No Delay

○ Set Transaction Rate

Interarrival Time: Absolute Value: 1000 milliseconds (ms)

● Real-World Delays

☑ Keying Time: Absolute Value: 300 milliseconds (ms)

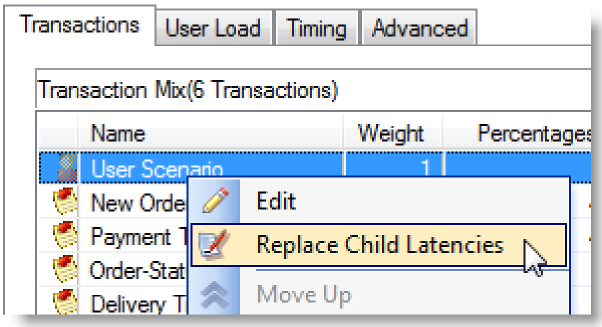
☑ Think Time: Absolute Value: 1000 milliseconds (ms)

Click here to see more about Latency.

4. Click **OK** to save your changes and close the dialog. The changes are applied to all the child transactions of the transaction mix. The changes are not inherited by the grandchild transactions. For example, if the transaction mix contains a user scenario, the latency values for the individual transactions in the user scenario remain unchanged. To change the latency values for transactions in the user scenario, right-click the user scenario and select **Replace Child Latencies**.

To replace child latencies at the User Scenario level

1. In the New/Edit Job wizard, select a transaction mix step and open the Transactions tab.
2. In the Transactions tab, right-click the user scenario and select **Replace Child Latencies**.



3. In the Replace Child Latencies dialog, modify latency options. See About Latency on page 143 for more information.

Use Global Search and Replace

Global Search and Replace allows you to search for and replace text from the selected item and all of its children.

To use Global Search/Replace

1. In the Jobs View, right-click a test you wish to search and replace.
2. Select **Global Search/Replace**. The Global Search and Replace dialog displays.
3. Enter the desired parameter in **Find What**.
4. Enter the desired parameter in **Replace With**.
5. Check Match Case or Match Whole word if desired.
6. Click **Replace**.

Save a Job as a Benchmark Factory Script

You can save a job as a Benchmark Factory script.

To save a job as a BMF Script

1. After creating and saving a job, select the job in the Jobs View pane and click **Save**, or select **File | Save**.
2. Browse to and select a location in which to save the file.
3. The job is saved as a .bfj file.

Copy a Test to a Replay, Mix, or Scalability Test

A test can be quickly converted from one test type to another by using the "copy" command. A workload test is an assembled test comprised of user scenarios and/or transactions. These tests can be run with multiple virtual users. A workload test can be one of three types: mix test, replay test, goal test, or scalability test.

In Benchmark Factory you can copy a:

- Replay test

  A Replay Test runs multiple transactions with each one running independently on a specified number of users. The test will run until the defined number of executions for each transaction or a specified time limit is reached.

- Mixed test

  A Mixed Workload test runs for a specified time at each predetermined user loads level. Each user will run a transaction mix based upon the weights defined on the transactions tab. For example, if a test has two transactions, A and B, with A having a weight of one and B having a weight of four, on average B will run four times for every time A is run once. The run order will be randomly generated for each user so they are not all running the same transaction simultaneously. That run order is used for that user each time the test is performed to ensure reproducible results.

- Goal test

  A goal test is used to find maximum throughput or response time goals. A transaction mix is executed at user load levels, determined by setting a beginning, ending, and interval value. When run, the specified goal criterion is evaluated at the end of each iteration and the test ends once the goal or maximum user load has been reached.

- Scalability

  A SQL Scalability test executes each transaction individually for each userload and timing period. For example, a test has two transactions, A and B, and two userloads of 10 and 20, with an iteration length of one minute. Transaction A would execute continually for one minute at userload 10, then B would do the same. Next A will run at userload 20, followed again by test B, for a total time of 4 minutes.

Copying a workload test creates a new test containing all compatible settings, user scenarios, and transactions from the original test.

To copy a workload test to a Replay Test, Mix Test, Goal Test, or Scalability test

1. Right-click a test in the Jobs View pane or in the New/Edit Job Wizard. A drop-down displays.
2. Select the desired type of test you want to copy to.
3. The test is created and displays in the Jobs View or New/Edit Job Wizard.

9

Benchmarks

Overview of Benchmark Testing

A benchmark is a performance test of hardware or software. A benchmark simulates real-world application workloads. These models of real applications can then be run against the system being evaluated. Models work better than the actual application and offer reproducible results. The real application, on the other hand, has too many variables that can change the results over several test sessions.

Realistic Expectations When Using Benchmarks

Industry standard benchmarks represent real application workloads, their run results depend on workload definition (user load, latency, etc.), as well as server configuration and tuning parameters. During the running of a benchmark, deadlock errors can occur. This is not an issue with Benchmark Factory, but more of an issue with the tuning of the workload as well as the system-under-test. For example, it would be unrealistic to expect no errors from a workload that runs 1000 users updating a single row on a small table. To further troubleshoot any tuning issues, we recommend the Quest Spotlight® products. Learn more at: http://www.quest.com/

What Benchmarks Measure

Benchmarks measure performance that analyze:

- Raw performance of a complete system
- Raw performance of a specific subsystem (disk, video, CPU, memory, etc.)
- Performance of a computer running a particular application
- Performance of applications running on a network
- Capacity of a system. This is often refereed to as capacity planning.

Provided Benchmarks

Benchmark Factory provides the following benchmark tests:

Benchmark	Benchmark Version (TPC)
AS3AP	--
Scalable Hardware	--
TPC-B	2.0
TPC-C	5.11
TPC-D	2.1

Benchmark	Benchmark Version (TPC)
TPC-E	1.12.0
TPC-H	2.14.2
Replication	--

AS3AP Benchmark

The AS3AP benchmark is an American National Standards Institute (ANSI) Structured Query Language (SQL) relational database benchmark. The AS3AP benchmark provides the following features:

- Tests database processing power
- Built-in scalability and portability that tests a broad range of database systems
- Minimizes effort in implementing and running benchmark tests
- Provides a uniform metric and straight-forward interpretation of benchmark results

Systems tested with the AS3AP benchmark must support common data types and provide a complete relational interface with basic integrity, consistency, and recovery mechanisms. The AS3AP tests systems ranging from a single-user microcomputer Database Management System (DBMS) to a high-performance parallel or distributed database.

To learn how to create an AS3AP benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Best Practices

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for AS3AP node to a job as needed. See Add a Create/Delete Benchmark Objects Test on page 132 for more information.
- Run the Load Scenario Wizard to create a new script containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'AS3AP' job will clean the environment.

Scaling Factor

The AS3AP benchmark scales by factor of 10.

Scalable Hardware Benchmark

The Scalable Hardware benchmark measures relational database systems. This benchmark is a subset of the AS3AP benchmark and tests the following:

- CPU
- Disk
- Network
- Any combination of the above three entities

To learn how to create a Scalable Hardware test in Benchmark Factory, see Create Industry Standard Benchmark Test.

How the Scalable Hardware Benchmark Works

The scale factor determines the amount of information initially loaded into the benchmark tables. For the Scalable Hardware benchmark, each scale factor represents one user accessing the system. Two tables are created in the database, and they are loaded with a varying number of rows.

For each virtual user, a separate set of data must be created. Therefore the scale factor used when loading the database should be the size of the maximum user load. For example, with user loads of 1, 5, and 10, a scale factor of 10 should be used.

Scaling Factor

The Scalable Hardware benchmark has a scaling factor of one.

Best Practices

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for the Scalable Hardware node to a job as needed.

- Run the Load Scenario Wizard to create a new script containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'Scalable Hardware" job will clean the environment.

TPC-B Benchmark

- Overview
- Certification of Transaction Processing Council (TPC) Testing Results
- Best Practices
- Scaling Factor

To learn how to create a TPC-B benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Overview

The Transaction Processing Council is an organization that establishes transaction processing and database benchmark standards. Find a complete overview and detailed explanation of the TPC-B Benchmark, at: http://www.tpc.org/tpcb/default.asp

Certification of Transaction Processing Council (TPC) Testing Results

Transaction Processing Council testing results cannot be published as certified unless the testing procedure is audited and approved by the TPC organization. If not certified, the testing results can be published as a "TPC-B like" test.

Best Practices

The following provides best practices for the TPC-B Benchmark.

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for TPC-B node to a job as needed.
- Run the Load Scenario Wizard to create a new script containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'TPC-B' job will clean the environment.

History Tables

The TPC-B benchmark is made up of only one transaction that updates three tables and inserts a record into a history table. Inserting one record into one history table limits testing performance. The Benchmark Factory properties page allows the user to set the number of history tables to create during a test. The best ratio of history tables to virtual users is based on database configuration and hardware. The number of history tables to use is determined by the tester.

Scaling Factor

The TPC-B benchmark scales by a factor of one.

TPC-C Benchmark

- Overview
- TPC-C Tables
- Certification of Transaction Processing Council (TPC) Testing Results
- Best Practices

To learn how to create a TPC-C benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Overview

Find a detailed overview of the TPC-C Benchmark at: http://www.tpc.org/tpcc/default.asp.

The TPC-C benchmark is an online transaction processing benchmark that simulates environments that have a number of terminal operators that send transactions to a database. This benchmark is focused on the concept of an order-entry type environment with transaction that include orders, payment recording, order status, and stock level monitoring. This benchmark portrays the activities of a wholesale supplier. However, the TPC-C is not limited to one particular business segment. It can represent numerous categories of a business that sell or distribute products and services.

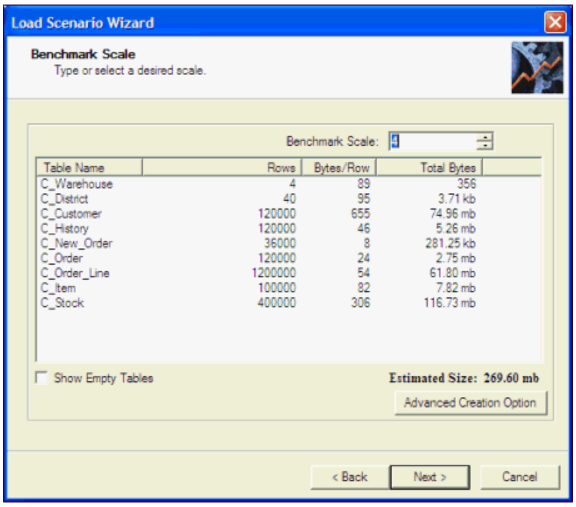
The TPC-C benchmark simulates a wholesale parts dealer operating out of warehouses. This Benchmark scales as a company, in theory, expands their business or number of facilities. As the TPC-C benchmark scales, so do the number components for the benchmark, for example, the sales districts and customers.

TPC-C Tables

The scale factor determines the amount of information initially loaded into the benchmark tables. For the TPC-C benchmark, each scale factor represents one warehouse as per TPC-C specification. The TPC-C benchmark involves a mix of five concurrent transactions of different types and complexity. The database is comprised of nine tables with a wide range of records.

A maximum of 10 users should be run against each warehouse. For example, user loads of 1, 5, and 10, set the scale to 1. If using other user load values, change the scale factor accordingly.

The TPC-C database consists of the following tables:



Benchmark Scale
Type or select a desired scale.

Benchmark Scale: 4

Table Name	Rows	Bytes/Row	Total Bytes
C_Warehouse	4	89	356
C_District	40	95	3.71 kb
C_Customer	120000	655	74.96 mb
C_History	120000	46	5.26 mb
C_New_Order	36000	8	281.25 kb
C_Order	120000	24	2.75 mb
C_Order_Line	1200000	54	61.80 mb
C_Item	100000	82	7.82 mb
C_Stock	400000	306	116.73 mb

☐ Show Empty Tables

Estimated Size: 269.60 mb

- Warehouse
- District
- Customer
- History
- New_Order
- Order
- Order_Line
- Item
- Stock

Certification of Transaction Processing Council (TPC) Testing Results

Transaction Processing Council testing results cannot be published as certified unless the testing procedure is audited and approved by the TPC organization. If not certified, the testing results can be published as a "TPC-C like" test.

Best Practices

The following provides best practices for the TPC-C Benchmark.

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for TPC-C node to a job as needed.
- Run the Load Scenario Wizard to create a new script containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'TPC-C' job will clean the environment.

TPC-D Benchmark

- Overview
- Certification of Transaction Processing Council (TPC) Testing Results
- Best Practices
- Scaling Factor

To learn how to create a TPC-D benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Overview

The Transaction Processing Council is an organization that establishes transaction processing and database benchmark standards. Find a complete overview and detailed explanation of the TPC-D Benchmark at: http://www.tpc.org/tpcd/default.asp

Certification of Transaction Processing Council (TPC) Testing Results

Transaction Processing Council testing results cannot be published as certified unless the testing procedure is audited and approved by the TPC organization. If not certified, the testing results can be published as a "TPC-D like" test.

Best Practices

The following provides best practices for the TPC-D Benchmark.

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for TPC-D node to a job as needed.
- Run the Load Scenario Wizard to create a new script containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'TPC-D' job will clean the environment.

Scaling Factor

The TPC-D benchmark scales by the following factors:

- 0.10
- 1.00
- 10.00
- 30.00
- 100.00
- 300.00

TPC-E Benchmark

- Overview
- Certification of Transaction Processing Council (TPC) Testing Results
- Best Practices
- Scaling Factor

To learn how to create a TPC-E benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Overview

The Transaction Processing Council is an organization that establishes transaction processing and database benchmark standards. Find a complete overview and detailed explanation of the TPC-E Benchmark at: http://www.tpc.org/tpce/default.asp

Certification of Transaction Processing Council (TPC) Testing Results

Transaction Processing Council testing results cannot be published as certified unless the testing procedure is audited and approved by the TPC organization. If not certified, the testing results can be published as a "TPC-E like" test.

Best Practices

The following provides best practices for the TPC-E Benchmark.

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Scaling Factor

The TPC-E benchmark scales by factor of 500.

TPC-H Benchmark

Review the following for information about the TPC-H benchmark.

- Overview
- Certification of Transaction Processing Council Testing Results
- Best Practices
- Stream Test

To learn how to create a TPC-H benchmark test in Benchmark Factory, see Create Industry Standard Benchmark Test.

Overview

The Transaction Processing Council is an organization that establishes transaction processing and database benchmark standards. Find a complete overview and detailed explanation of the TPC-H Benchmark at: http://www.tpc.org/tpch/default.asp.

Certification of Transaction Processing Council Testing Results

Transaction Processing Council testing results cannot be published as certified unless the testing procedure is audited and approved by the TPC organization. If not certified, the testing results can be published as a "TPC-H like" test.

Best Practices

The following provides best practices for the TPC-H Benchmark.

Load-testing against production servers

Do not load-test against a production server if possible. Load-testing and benchmarking on a production server significantly degrades performance. In some cases, load-testing can cause a server to fail. However, if testing against a production server, take the following precautions:

- Perform the testing when no other users are on the system and no automated processes are running. Users and automated processes can adversely affect testing results
- Have a recovery plan and backup all data prior to testing
- Determine how long it will take to restore a production server if it went down during load-testing
- Perform manual testing. Manual testing ensures that no unexpected outside activity takes place during the testing process

Reinitialize the Database

To reinitialize a testing database, run a job containing a Benchmark Object node.

There are two ways to create a Benchmark Object node:

- Run the Benchmark Object Wizard to add the Delete Benchmark objects for TPC-H node to a job as needed.
- Run the New Job Wizard to create a new job containing the Create Objects for the Benchmark, Associated Load Scenarios for the Benchmark, and Delete Objects for the Benchmark. Running the delete objects for 'TPC-H' job will clean the environment.

Stream Test

An option when creating a TPC-H workload is to include the TPC-H Stream Test, which is the multi-user version of the Power Test. The Steam Test, per specification, should maintain the following relationship between the scale factor and the number of Streams.

Scale Factor	Streams
1	2
10	3
30	4
100	5
300	6
1000	7
3000	8
10000	9
30000	10

About the Replication Benchmark Test

The Replication benchmark test is a unique test developed by Benchmark Factory for testing replication in applications such as SharePlex.

This test allows you to create customized database objects, as well as a customized workload, in order to produce a repeatable database change rate and history (for example, Oracle redo logs) that can be used to test replication strategies. This test can also be used to create a model of your database and workload in order to help demonstrate replication performance.

While this benchmark was designed for testing and evaluating replication strategies and functionality in SharePlex, it can also be used to test other database replication products as well.

The following options can be customized in the Replication benchmark test:

- The number of tables in the test
- The number of columns per table and the column types (data types)
- The percentage of each column type (data type) in all tables, for example, 40% INT, 20% VARCHAR (255), etc.
- The percentage of each statement type (insert, update, delete), as well as the number of statements per commit (transaction)

See Create a Replication Benchmark Test to learn how to create a new Replication test.

10

How Do I...

About Benchmark Factory Connections

Benchmark Factory allows you to create the following connections:

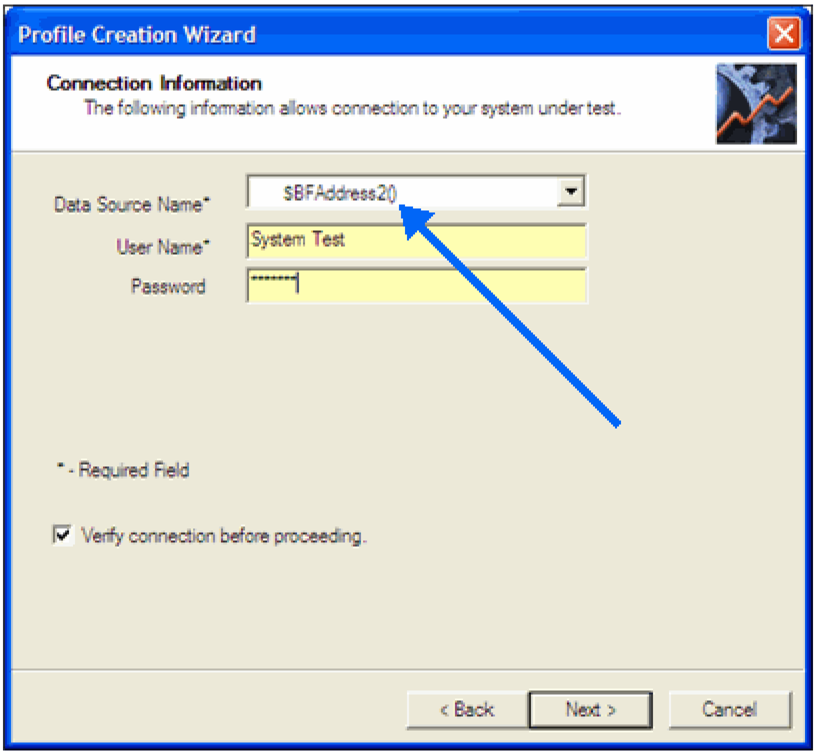
- IBM DB2CLI
- Microsoft SQL Server
- MySQL (ODBC)
- ODBC
- Oracle
- SAP

Adding BFScript to the Connection Dialog

You can add BFScripts to the Benchmark Factory Data Source Names, allowing you to have individual agents test against a different server. BFScripts have been enabled on the User name and Password fields.

To add BFScripts to the Data Source Name

1. Click inside **Data Source Name**.
2. Type the desired script name.



$BFAddress2()

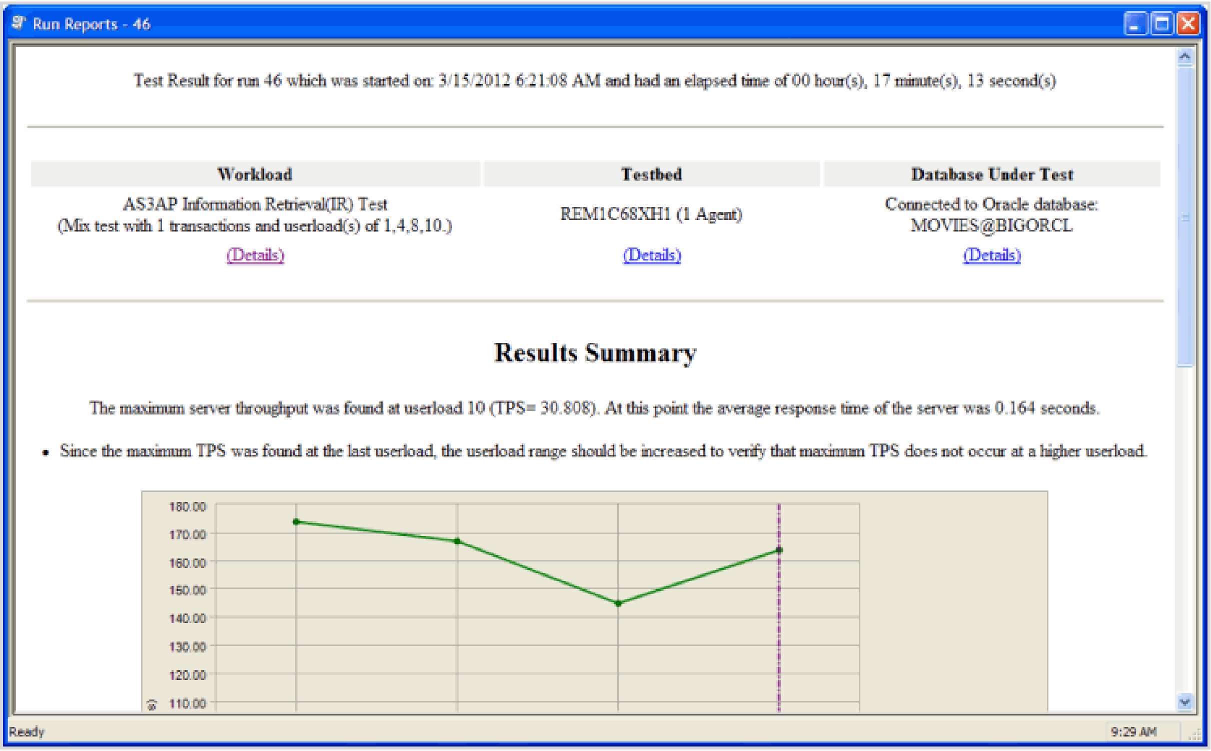
System Test

****

Run Reports

Benchmark Factory Run Reports is a separate executable that provides a comprehensive and detailed collection of database load testing results. With Benchmark Factory you can drill down into a database to view a wide array of information and statistics that gives you accurate insight into database performance. Run Reports Viewer allows you to access Benchmark Factory load testing results.

Note: Three instances of Run Reports can be viewed at one time.



Test Result for run 46 which was started on: 3/15/2012 6:21:08 AM and had an elapsed time of 00 hour(s), 17 minute(s), 13 second(s)

Workload	Testbed	Database Under Test
AS3AP Information Retrieval(IR) Test (Mix test with 1 transactions and userload(s) of 1,4,8,10.) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

Results Summary

The maximum server throughput was found at userload 10 (TPS= 30.808). At this point the average response time of the server was 0.164 seconds.

- Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

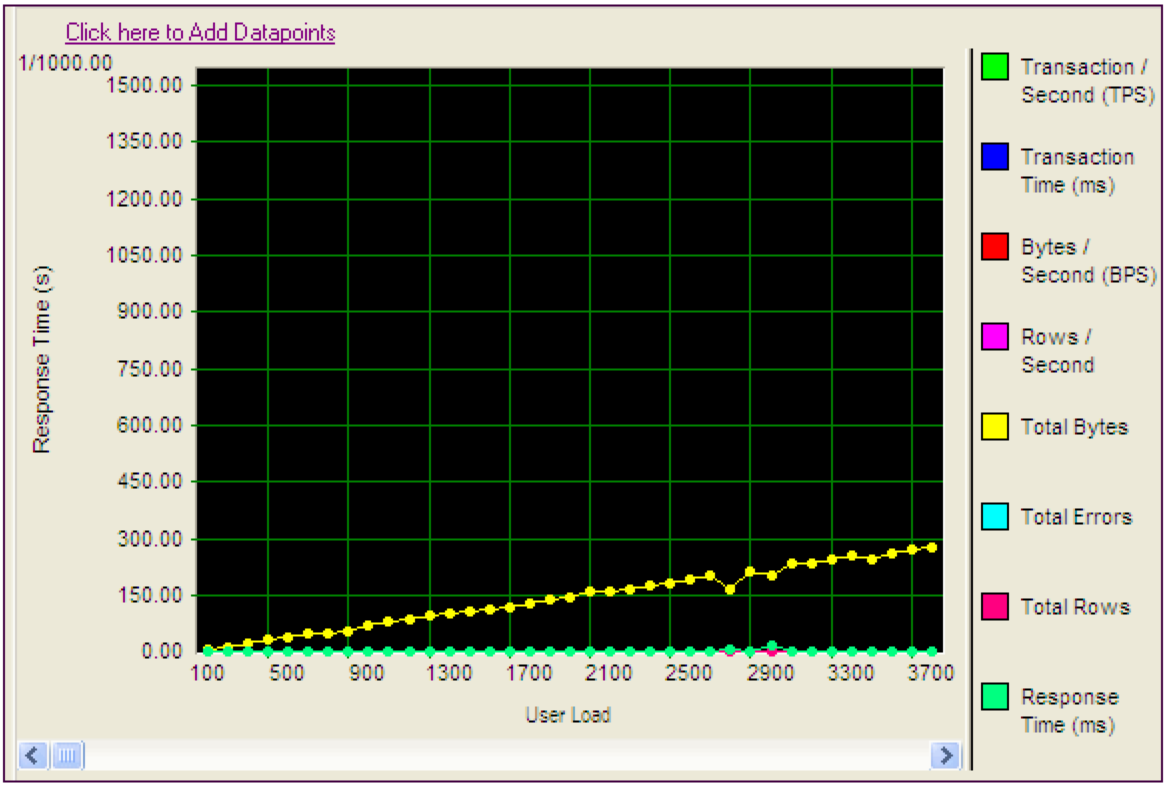
Cutting and Pasting of Run Reports

From Benchmark Factory run reports you can cut and paste test results into the Clipboard.

Testing Server Throughput

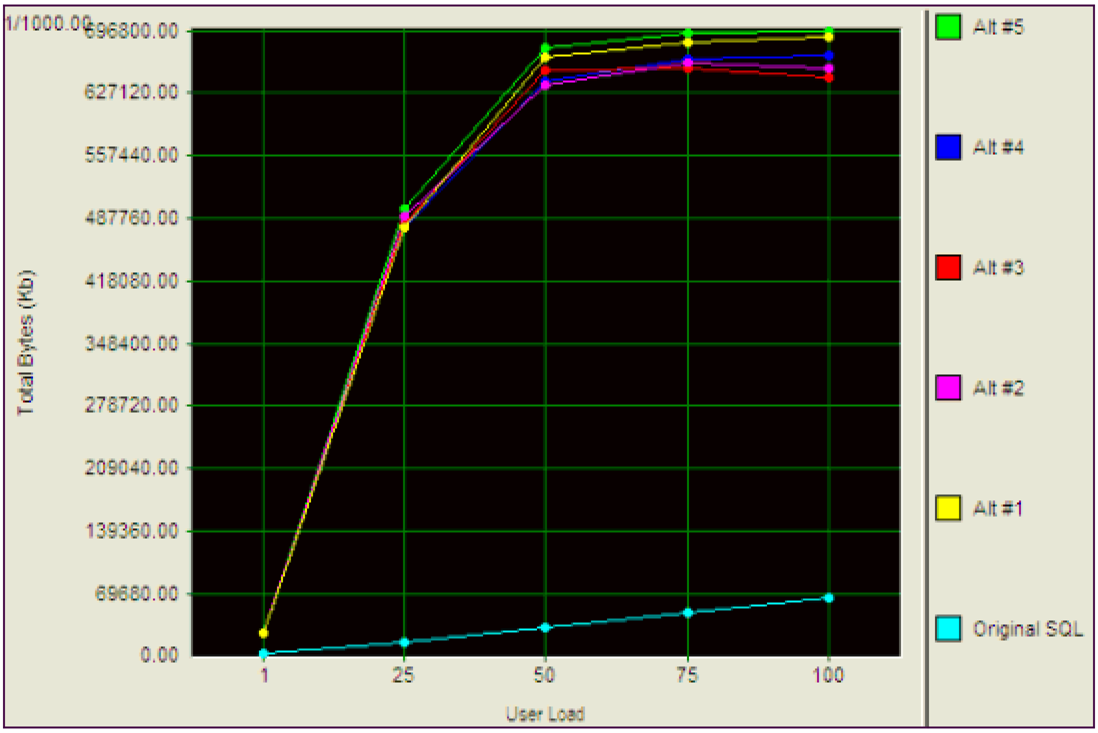
Server throughput is the measurement of database capacity. Benchmark Factory allows you to view a variety of server throughput statistics that include:

- Transaction/Second (TPS)
- Transaction Time
- Bytes/Second (BPS)
- Rows/Second (RPS)
- Total Bytes
- Total Errors
- Total Rows
- Response Time



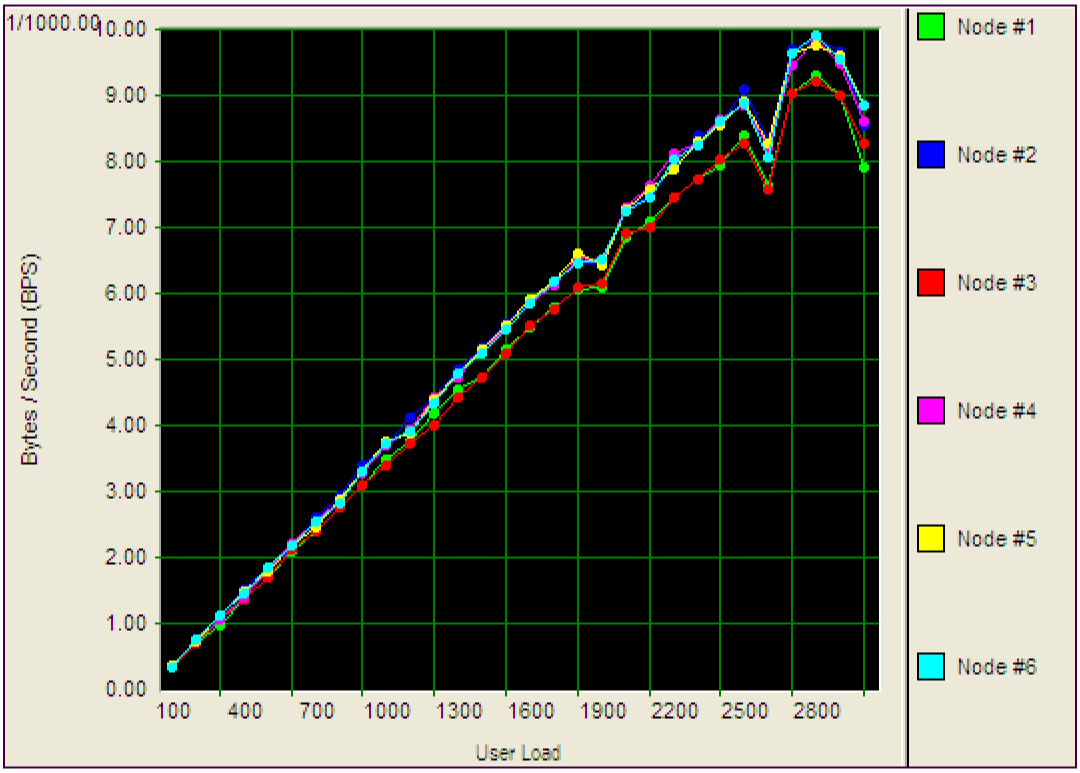
Testing SQL Scalability

The Benchmark Factory transaction comparison graph provides users with a comparison of the individual transaction performance as they compare to each other. This is most useful when analyzing a SQL Scalability test.



Testing Real Application Clusters

Benchmark Factory allows you to load test Oracle Real Application Clusters (RAC) to determine system capacity. The cluster performance graph allows users to view the performance of individual nodes.



Add a Create/Delete Benchmark Objects Test

The Create/Delete Benchmark Objects step creates or clears the database objects (tables/indexes, etc.) needed to run industry standard benchmarks. The step is included when you create an Industry Standard Benchmark Test. You can also add the step to an existing workload through the Edit Job Wizard.

Note: To create, delete, and modify database tables/indexes you **must** have the user login rights required.

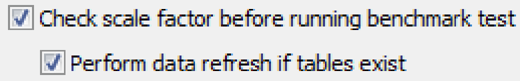
To add a Create/Delete Benchmark Objects Test to a Workload

1. Use one of the following methods to open the Job Wizard:
   - Select **File | New** to open the New Job Wizard.
   - Select an existing job and click **Edit Job**.
2. Click **Workload** in the left pane. If this is an existing job, click **Add Another Test/Step**
3. On the Workload page, select **Create/Delete Benchmark Objects Test** from the drop-down list.
4. Then select the benchmark type from the **Benchmark Type** drop-down list.

5. Select one of the following options:

    - **Create Benchmark Objects**—Creates and initializes benchmark objects (tables, indexes, etc.).
    - **Delete Benchmark Objects**—Deletes existing benchmark objects (tables, indexes, etc.).

6. If you selected **Create Benchmark Objects**, select one of the following options:

Create/Load objects if objects don't exist (no backup sets created)	If objects do not exist, Benchmark Factory creates the objects and loads data. If you select this option, Benchmark Factory does not create backup tables.
Delete benchmark objects after completion of job	Benchmark Factory deletes objects after a job is finished.
Refresh data using backup sets	When backup sets exist, Benchmark Factory reloads data and recreates indexes. If backup sets do not exist, Benchmark Factory creates objects and loads data. Then Benchmark Factory creates the backup tables.
Recreate objects and reload data every execution	Benchmark Factory always deletes the existing objects and then recreates/reloads for each job execution. This allows you to ensure that the dataset is always in initialized status.
Refresh data using inserts	If data does not exist, Benchmark Factory uses bulk insert to load data. If the following Benchmark Settings are selected,  ☑ Check scale factor before running benchmark test ☑ Perform data refresh if tables exist and the Benchmark Scale factor of the existing tables does not match the new requirement, the process of inserting additional data or deleting data my run long, depending on the size of the existing data and number of indexes. **Note:** If database does not support bulk insert, insert is used.

7. Click the **Import Test** button at the bottom of the page.

8. **Summary.** The test is added to the workload and the Summary page opens. Review the following editable options for this test/step.

    - **Scale**—Click the **Scale** link to modify the scale factor. The Scale page opens. In the **Benchmark Scale** field, select a new scale factor. See Benchmark Scale Factor on page 134 for more information.

    Note: The test is added to the top of the workload list. To move the test down (or up) in the list, right-click the test and select **Move Down** (or **Move Up**).

9. **Test Options.** To modify test options for the Create Objects test/step, select the Test Options tab. Review the following for more information:

    - **Scale**—To modify the scale factor. See Benchmark Scale Factor on page 134 for more information.
    - **Options**—To modify options for object creation and retention. See Options Tab (Create Objects Step) on page 95 for more information.

- **Custom Creation SQL**—To use custom SQL to create objects. See Customize SQL for Creating Objects on page 96 for more information.

ℹ **Tip:** You can specify a default setting for Object/Data Retention in **Edit | Settings | Benchmarks**.

Benchmark Scale Factor

When you create an Industry Standard Benchmark test or add a Create Benchmark Objects step, Benchmark Factory allows you to adjust the Benchmark scale factor. The scale factor increases the size of a database during the testing process allowing realistic testing.

By increasing the scale factor, the number of rows added to the tables increases, allowing for the use of larger user loads. This places a greater stress on the system-under-test. During load testing, you can run various tests with different Benchmark Scale factors to ensure your database performs at varying user-load levels.

You can adjust the Benchmark Scale factor when creating a new Industry Standard Benchmark Test or when adding a Create Benchmark Objects step.

To change the Benchmark Scale factor

1. In the New Job Wizard or Edit Job Wizard, select the Create Objects step (in the Workload) for which you want to adjust the scale.
2. Select the **Test Options** tab and then the **Scale** tab.
3. In the **Benchmark Scale** field, specify a scale factor.

   Valid benchmark scales are dependent upon the benchmark selected. For example, the AS3AP benchmark scales by a factor of 10 with valid scales of "1, 10, 20, 30, etc." The TPC-C benchmark scales by a factory of 1, with valid scales of "1, 2, 3, 4, etc." Refer to the specific benchmark standard for more information.

   (Replication test only) For a Replication test, each increment in scale factor adds 1000 rows to a table.
4. (Optional) Instead of specifying a scale factor, you can specify a **Benchmark Size**. The Benchmark Scale factor will re-adjust accordingly.
5. Finish specifying options for your test, or click **Save/Close** to save your modifications and close the wizard.

Benchmark Scale Factor and User Load Relationship

The TPC-C and the Scalable Hardware standard benchmark tests have a scale factor / user load relationship.

- For the TPC-C benchmark, the ratio is 10 virtual users per scale factor.
- For the Scalable Hardware benchmark, the scale factor is also the number of users that will return data from their queries.

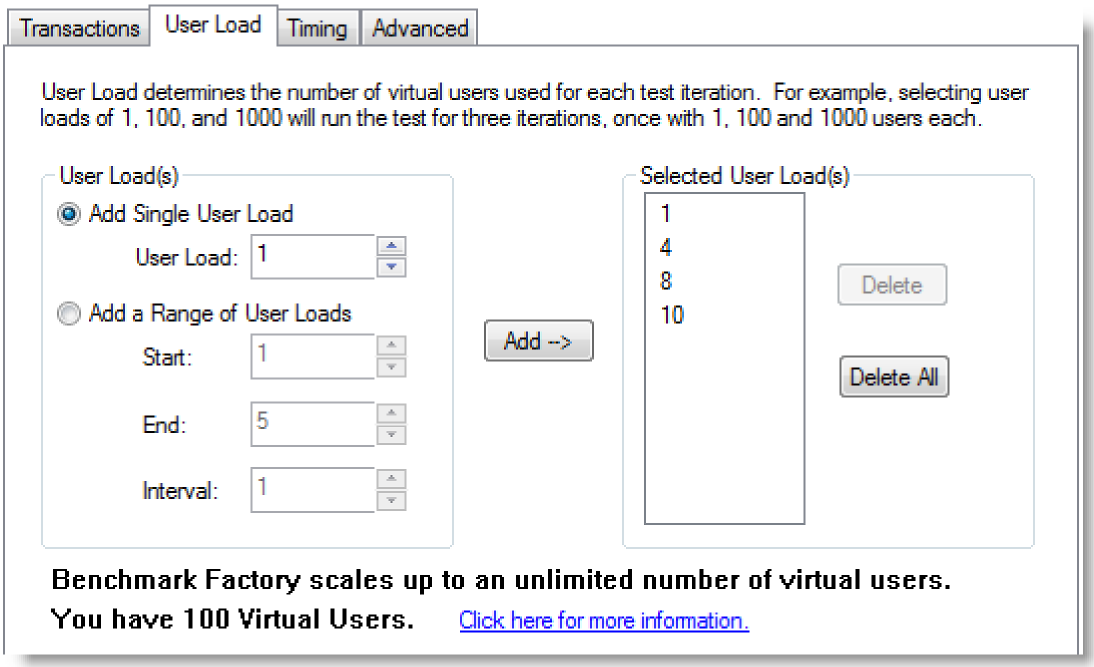
If you attempt to create a benchmark test using a scale factor / user load ratio outside of these specifications, Benchmark Factory warns you.

ℹ **Note:** You can set a default Benchmark Scale factor to use for each standard benchmark test. Go to **Edit | Settings | Benchmarks | *benchmark type***.

Specify User Load

The User Load tab of the New Job or Edit Job Wizard allows you to specify the number of virtual users for each iteration of a test.

Note: To set default values for User Load, go to **Edit | Settings | User Load**. If you modify the default settings, the changes will apply to new jobs only, not to existing jobs.



User Load determines the number of virtual users used for each test iteration. For example, selecting user loads of 1, 100, and 1000 will run the test for three iterations, once with 1, 100 and 1000 users each.

User Load(s)

● Add Single User Load

User Load: 1

○ Add a Range of User Loads

Start: 1

End: 5

Interval: 1

Selected User Load(s)

1
4
8
10

Benchmark Factory scales up to an unlimited number of virtual users.

You have 100 Virtual Users. Click here for more information.

Section	Description
User Load	**Add Single User Load**—Specify a single user load, then click **Add** to add it to the selected user loads. **Add a Range of User Loads**—Specify a range, then click **Add** to add the range to the selected user loads.

Execute External File

You might want to add an executable file (.bat or .exe) as part of a job. This gives you the flexibility to set up your database before running a workload. For example, you could run a .bat file to set up a prepared SQL statement before running a workload.

To add an Execute File step to a workload

1. In the New Job or Edit Job Wizard, click **Workload** in the left pane.
2. On the workload page, select *Execute External File* from the drop-down list.
3. Enter the path or browse to and select a file to execute.
4. To add a timeout, select the **Enforce Timeout** option and enter the number of seconds for Benchmark Factory to wait for the execution to complete before continuing the job.

   You can specify a default setting for this option in **Edit | Settings | Execute File**.

BFScript Wizard

Note: Controls with yellow background can be scripted.

Benchmark Factory scripting allows you to randomize the load testing process. Scripting can be utilized by selecting the SQL Statement node and inserting the script in the properties view. A field has scripting capabilities if its right-click menu contains the Script Wizard option.

The Script Wizard is a quick and easy way to use Benchmark Factory scripts. The Script Wizard consists of a script category and a script function. Each script function has a short description included and if applicable, its parameters.

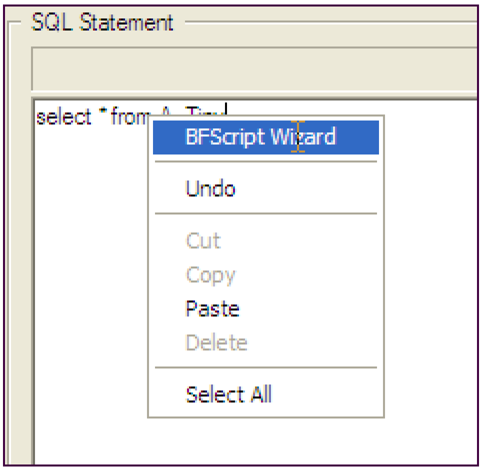
There are two features within Benchmark Factory that assist you when using scripting capabilities:

- Script Wizard
- Script Assist

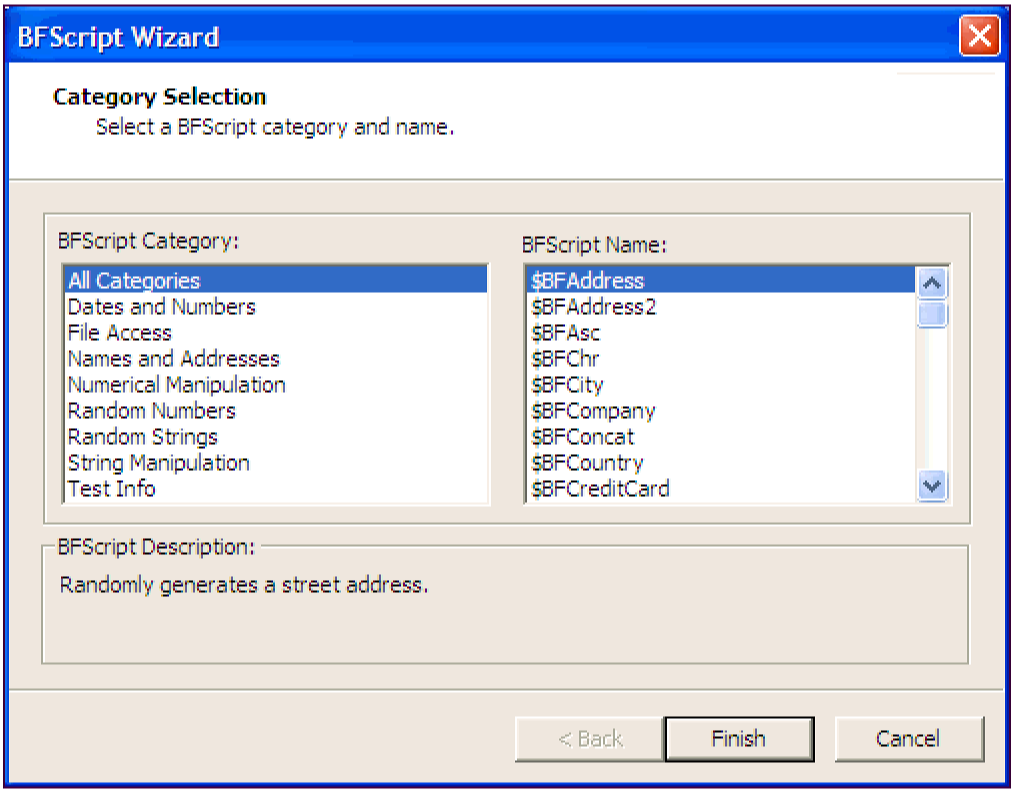
Click here for an overview of $BFScripts.

Script Wizard

1. Right-click the edit control. A drop-down display.



2. Select **BFScript Wizard** from the drop-down. The Script Wizard displays.



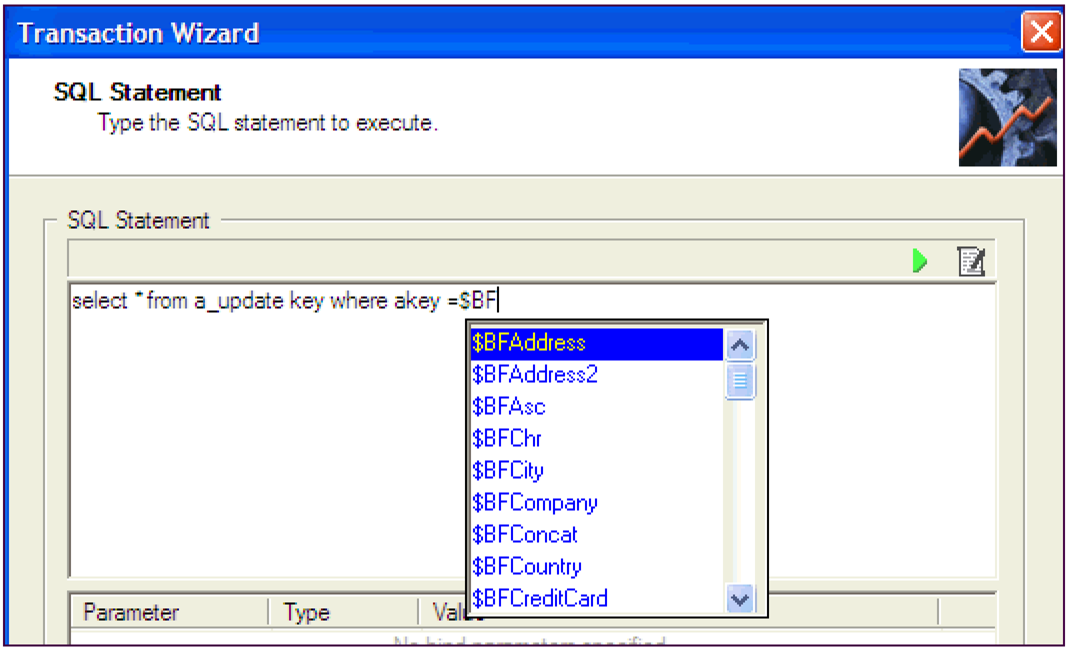
3. Select the desired script.
4. Some scripts require further parameters settings. For a complete overview of individual script settings, please refer to Scripts Overview.

   If scripts require parameters:

1. Click **Next.**
2. Enter the required parameters.
5. Click **Finish**. The script and parameters display in the Script Wizard.

Script Assist

1. Type in the desired SQL statement. When $BF is typed the Script Assist is activated allowing you to insert the desired script.



select * from a_update key where akey =$BF

2. Select the desired script.
3. Press **Enter**. The script displays.
4. Type in parameters as required.

Copy a Test to a Replay, Mix, or Scalability Test

A test can be quickly converted from one test type to another by using the "copy" command. A workload test is an assembled test comprised of user scenarios and/or transactions. These tests can be run with multiple virtual users. A workload test can be one of three types: mix test, replay test, goal test, or scalability test.

In Benchmark Factory you can copy a:

- Replay test

  A Replay Test runs multiple transactions with each one running independently on a specified number of users. The test will run until the defined number of executions for each transaction or a specified time limit is reached.

- Mixed test

  A Mixed Workload test runs for a specified time at each predetermined user loads level. Each user will run a transaction mix based upon the weights defined on the transactions tab. For example, if a test has two transactions, A and B, with A having a weight of one and B having a weight of four, on average B will run four times for every time A is run once. The run order will be randomly generated for each user so they are not all running the same transaction simultaneously. That run order is used for that user each time the test is performed to ensure reproducible results.

- Goal test

  A goal test is used to find maximum throughput or response time goals. A transaction mix is executed at user load levels, determined by setting a beginning, ending, and interval value. When run, the specified goal criterion is evaluated at the end of each iteration and the test ends once the goal or maximum user load has been reached.

- Scalability

  A SQL Scalability test executes each transaction individually for each userload and timing period. For example, a test has two transactions, A and B, and two userloads of 10 and 20, with an iteration length of one minute. Transaction A would execute continually for one minute at userload 10, then B would do the same. Next A will run at userload 20, followed again by test B, for a total time of 4 minutes.

Copying a workload test creates a new test containing all compatible settings, user scenarios, and transactions from the original test.

To copy a workload test to a Replay Test, Mix Test, Goal Test, or Scalability test

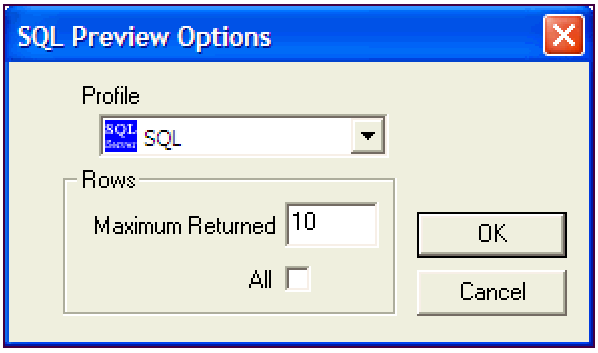
1. Right-click a test in the Jobs View pane or in the New/Edit Job Wizard. A drop-down displays.
2. Select the desired type of test you want to copy to.
3. The test is created and displays in the Jobs View or New/Edit Job Wizard.

Run SQL Preview

SQL Preview allows you to preview the result set of an individual SQL statement when adding or editing a SQL transaction.

To run SQL Preview

1. In the SQL Statement tab, click ▶.
2. Select a connection from the drop-down list.



SQL Preview Options

Profile: SQL

Rows

Maximum Returned: 10

All ☐

4. In the **Maximum Returned** field, enter the maximum number of rows to preview, or select the **All** option to preview all rows.
5. Click **OK**. The SQL Preview window opens.
6. Review the information.
7. Click **Close**.

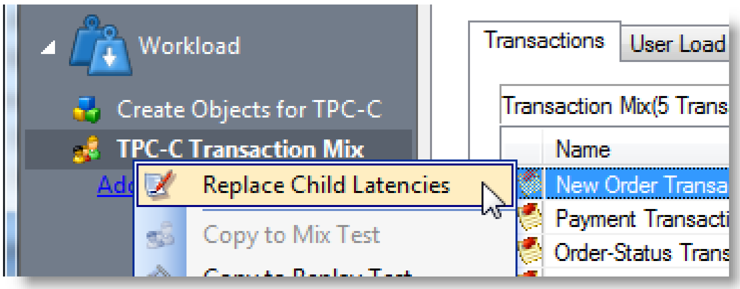
Replace Child Latencies

Latency is delay added to the execution of a transaction to control of how fast transactions are submitted to the system-under-test. You can use the **Replace child latencies** dialog to quickly modify latency for all the direct children of a test. For example, when an Oracle trace file is imported, the latencies are automatically set to replay the transactions at their original rate, but changing these to a set value can replay the transactions in a shorter period of time.

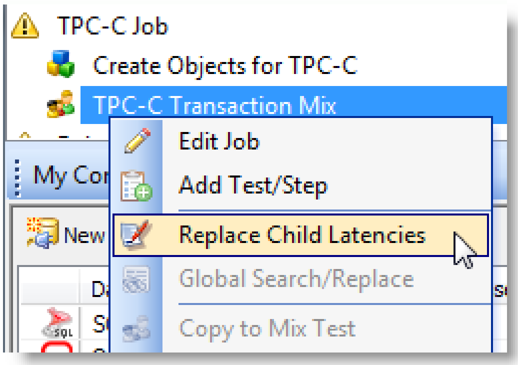
To replace child latencies at the test level

1. Use one to the following methods to open the Replace Child Latencies dialog:

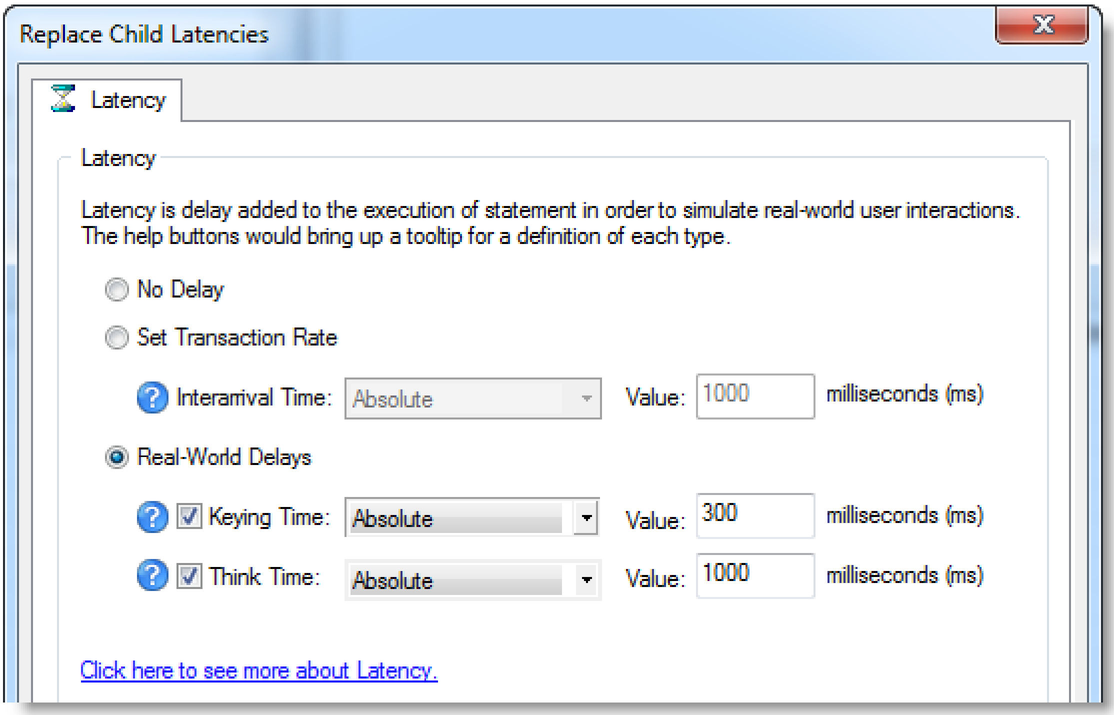
   - In the New/Edit Job Wizard, right-click a test and select **Replace Child Latencies**.



   - In the Jobs View pane, right-click a test and select **Replace Child Latencies**.



2. In the Replace Child Latencies dialog, modify latency options. See About Latency on page 143 for more information.



Latency

Latency is delay added to the execution of statement in order to simulate real-world user interactions. The help buttons would bring up a tooltip for a definition of each type.

○ No Delay

○ Set Transaction Rate

Interarrival Time: Absolute Value: 1000 milliseconds (ms)

● Real-World Delays

☑ Keying Time: Absolute Value: 300 milliseconds (ms)

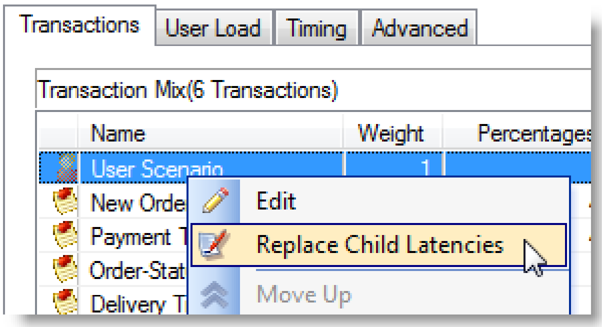
☑ Think Time: Absolute Value: 1000 milliseconds (ms)

Click here to see more about Latency.

4. Click **OK** to save your changes and close the dialog. The changes are applied to all the child transactions of the transaction mix. The changes are not inherited by the grandchild transactions. For example, if the transaction mix contains a user scenario, the latency values for the individual transactions in the user scenario remain unchanged. To change the latency values for transactions in the user scenario, right-click the user scenario and select **Replace Child Latencies**.

To replace child latencies at the User Scenario level

1. In the New/Edit Job wizard, select a transaction mix step and open the Transactions tab.
2. In the Transactions tab, right-click the user scenario and select **Replace Child Latencies**.



3. In the Replace Child Latencies dialog, modify latency options. See About Latency on page 143 for more information.

Use Global Search and Replace

Global Search and Replace allows you to search for and replace text from the selected item and all of its children.

To use Global Search/Replace

1. In the Jobs View, right-click a test you wish to search and replace.
2. Select **Global Search/Replace**. The Global Search and Replace dialog displays.
3. Enter the desired parameter in **Find What**.
4. Enter the desired parameter in **Replace With**.
5. Check Match Case or Match Whole word if desired.
6. Click **Replace**.

Creating an ODBC Trace File

To create an ODBC trace file

1. Click the **Start** button from the Windows taskbar.
2. Select **Settings | Control Panel**. The Control Panel window displays.
3. Double-click on **Administrative Tools**. The Administrative Tools window displays.
4. Launch **Data Source (ODBC)**. The ODBC Data Source Administrator window displays.
5. Click the **Tracing** tab.
6. Click the **Start Tracing Now** button. The Start Tracing Now button changes to Stop Tracing Now button.

7. Run the ODBC transactions to be traced.

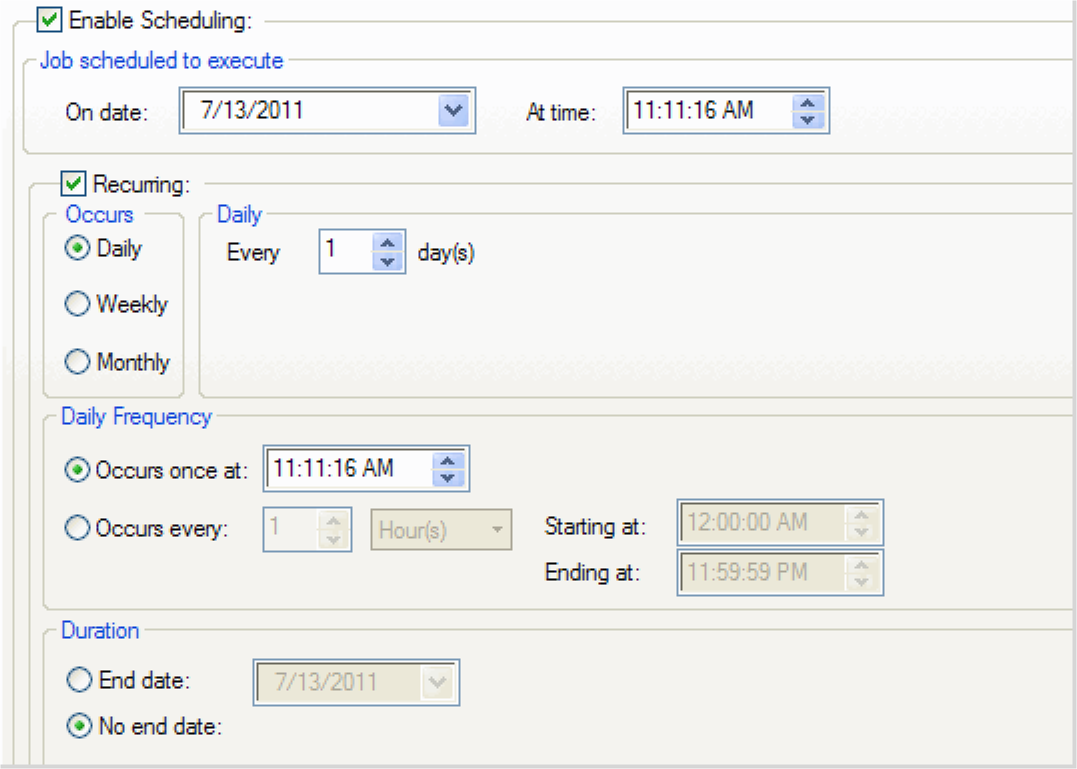
8. When the transactions complete, click on the Stop Tracing Now button.

9. Click **OK**.

Schedule a Job

Benchmark Factory allows you to schedule a job to run at a future time. Scheduling allows you to set predetermined dates, recurrences, daily frequencies, and durations of jobs.

Note: When using the recurring option, use Ready Job in place of Run Job.



☑ Enable Scheduling:

Job scheduled to execute

On date: 7/13/2011 At time: 11:11:16 AM

☑ Recurring:

Occurs

● Daily

○ Weekly

○ Monthly

Daily

Every 1 day(s)

Daily Frequency

● Occurs once at: 11:11:16 AM

○ Occurs every: 1 Hour(s) Starting at: 12:00:00 AM

Ending at: 11:59:59 PM

Duration

○ End date: 7/13/2011

● No end date:

To schedule a job

1. In the New/Edit Job Wizard, select a test under **Workload**.
2. Select the **Job Setup** tab.
3. Select the **Schedule** tab.
4. Select the **Enable Scheduling** checkbox.
5. To run the job once, select a date and time to execute the job.
6. To run the job on a recurring basis, select a start date and time, and select the **Recurring** check box.
    a. Then specify a frequency.
    b. To run the job multiple times per daily, specify options under Daily Frequency.
7. Select an end date, or select the **No end date** checkbox.
8. Click **Save/Close** to save your options and close the wizard.

About Latency

Latency is delay added to the execution of a transaction to control of how fast transactions are submitted to the system-under-test. It is used to either make the transaction execution rate more like real-world executions or to control the transaction rate. This delay can be added to the beginning and/or end of a transaction execution.

You can modify latency values for a transaction by editing the transaction in the Transactions tab of the New/Edit Job wizard. In addition, you can specify latency values at the time you add transactions/scenarios to a test using the Transactions tab.

Benchmark Factory allows you to set default latency values for the transactions you add.

To modify latency values

1. Open the New Job wizard for a new job or the Edit Job wizard for an existing job.
2. Select the test (or the step) under **Workload** that contains the transactions.
3. Select the **Test Options** tab. Then select the **Transactions** tab.
4. Select the transaction or scenario for which you want modify latency and click **Edit**.
5. In the dialog that opens, select the **Latency** tab. For a scenario, to modify latency for individual transactions, select a transaction and click **Edit**, and then select the Latency tab.
6. Modify latency options. To specify no latency, select **No Delay**. To add latency, select one of the other options. Review the following for additional information:

No Delay	Select to specify no latency.
Transaction Rate	Select to set a transaction rate by specifying the interval between successive transactions arriving at the server. **Interarrival Time**—Select one of the following: • Absolute—Select to specify a fixed interarrival time. Then enter a value. • Distribution models—To use a distribution model for determining interarrival time, select a model and enter the requested value.
Real-World Delays	Use this option to simulate real-world processing by inserting a delay before and/or after each transaction execution. For each delay type, you can insert a fixed time by selecting *Absolute*, or you can select one of the distribution models for specifying time. **Keying Time**—Inserts a delay before execution to simulate the amount of time spent performing data entry. **Think Time**—Inserts a delay after execution to simulate the amount of time "to think" about the results of the previous transaction.

7. Click **OK** to save your changes and close the dialog.
8. Then click **Save/Close** in the wizard to save your changes.

Why Use Latency?

Real-World Delays: Add Keying Time and Think Time to create a workload that accurately simulates real-world users, as opposed to a workload that submits transactions as fast as possible. Using real-world delays allows for easier interpretation of the results. If you do not add delay, it might be difficult to equate *N virtual users running with no delay* to an accurate number of real users.

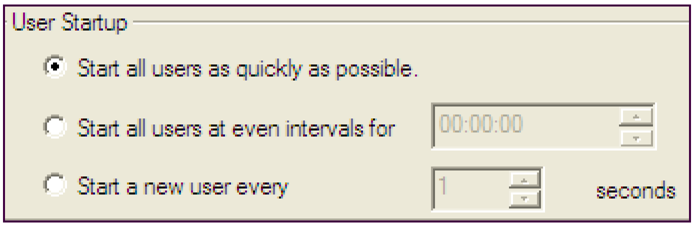
Interarrival Time: Specify Interarrival Time to create a workload in which transactions arrive at the system-under-test at a specific interval. This method allows you to set the timing relationship between executing transactions and is commonly used when replaying a captured workload.

Stagger User Startup

When running a load test, you have the option of staggering user startup. This allows you to adjust startup loads if, for example, you are having issues with overloading of servers when all of the users attempt to start at the same time.

To stagger user startup

1. In the Edit/New Job Wizard, select a test under **Workload**.
2. Select the **Timing** tab (under Test Options tab).
3. In the User Startup options, select the desired parameters.



User Startup

● Start all users as quickly as possible.

○ Start all users at even intervals for 00:00:00

○ Start a new user every 1 seconds

- **Start all users as quickly as possible**: Starts all users immediately after a test begins.
- **Start all users at even intervals for**: Sets the amount of time in which to start all users at even intervals. The interval duration is equal to this value divided by the number of users.
- **Start a new user every**: Starts a new user, then waits the entered number of seconds before starting the next user.

Running Benchmark Factory with Multiple Agents

You can install agents on multiple agent machines in order to run tests using multiple agents. To configure Benchmark Factory with multiple agents you must:

- Load the proper client libraries (i.e. Oracle, SQL server, etc.) on the Benchmark Factory Console and Agent Machine.
- Install Benchmark Factory on the local and agent machines.

Install Multiple Agents

For best results, install remote agents through your Benchmark Factory console. You can install remote agents from the Settings dialog (Edit | Settings | Agent) or from the Agent page of the New/Edit Job Wizard. See Install Remote Agents on page 38 for more information.

You can also install agents manually on the agent machine. If you do, you must configure the agent on the agent machine (see below). Then, in your local Benchmark Factory, you must set up the agent to register it with the console. See Agent Settings on page 160 for more information.

Configure Agent Machine

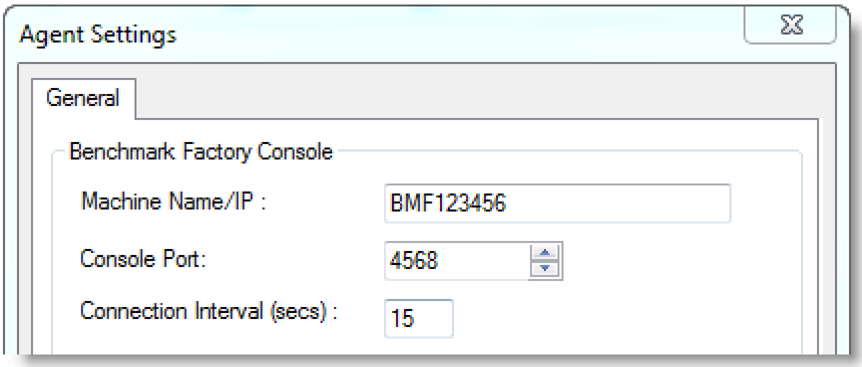
If you set up an agent machine by installing Benchmark Factory manually, you must configure the agent on the agent machine.

Each Benchmark Factory Agent must be configured with the address of the Benchmark Factory Console. Each Agent sends load testing results back to the Benchmark Factory Console.

If you use only the agent installed locally on the console machine, make sure your local agent is configured with the IP address (name) of your local machine.

To configure Benchmark Factory Agents

1. On each agent machine, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe** to open the Agent dialog.
2. Select **Options | Settings**.
3. In **Machine Name/IP** field, enter the address of the Benchmark Factory console machine. In the **Console Port** field, enter the port for the Benchmark Factory console.



Agent Settings

General

Benchmark Factory Console

Machine Name/IP : BMF123456

Console Port: 4568

Connection Interval (secs) : 15

4. Click **OK**.
5. Repeat this procedure for each Agent machine

Set Up the Agents

To run a job using multiple agent machines, after the agents are installed and configured, you must set up (or add) each remote agent to the Benchmark Factory console.

- If the agent was installed manually, you can set up the agent in **Edit | Settings | Agent**. See Agent Settings on page 160 for more information.
- If the agent was installed remotely through the Benchmark Factory console, the agent is automatically set up.

Running a Job With Multiple Agents

To run a job with multiple agents

1. Open the New Job Wizard and create a workload.
2. Click **Agent** in the left panel of the New Job Wizard.
   - For an existing job, open the job in the Edit Job Wizard and click **Agent** in the left pane.

3. From the list, select the agents to use. The list displays agents you have previously set up in Benchmark Factory.
    - To add an agent, click **Setup New User Agent**. See Agent Settings on page 160 for more information.
4. As the job runs, all connected Agents will display in the Agent view/pane of the Benchmark Factory console.

Note: When you run a job using one or more local agents, if Agent utilization of resources on the local machine is too high, errors could occur.

View Agents from Benchmark Factory Console

When you run multiple Agents for load testing, you can view the Agents results as they run from the Benchmark Factory Console.

To view the Agents from the Benchmark Factory Console

1. From the Benchmark Factory main menu, select **View | Agent**. The Agent view displays. Statistics from All Agents connected to the Benchmark Factory Console and running are displayed here.

SQL Scalability Testing from Toad

Note: To run SQL Scalability testing from Toad you must have Benchmark Factory installed. The Benchmark Factory console does not have to be open to run SQL Scalability testing.

Overview

The SQL Scalability test allows you to execute SQL statements, letting users spot potential issues not seen with a single execution. Users can run variations of a SQL statement generated by SQL Tuning in order to find the SQL that will perform the best under a load test.

Creating a SQL Scalability load scenario requires the following steps.

1. Creating the SQL tuning connection
2. Entering the desired SQL statement
3. Running the statement using the Benchmark Factory SQL Scalability testing

Creating the SQL tuning connection

To create the SQL Turning connection

1. In Quest Central or Toad for Oracle, connect to an Oracle or SQL Server database.
2. From the Quest Central main toolbar select **Tools|SQL Tuning** to launch.

Entering the desired SQL statement

To enter the desired SQL statement

1. Enter the desired SQL statement in **SQL Text**.
2. Click the **Optimize Statement** icon. Clicking this icon executes multiple syntax transformation rules to produce a list of semantically equivalent SQL statements.

Running the SQL statement using the in Benchmark Factory SQL Scalability Testing

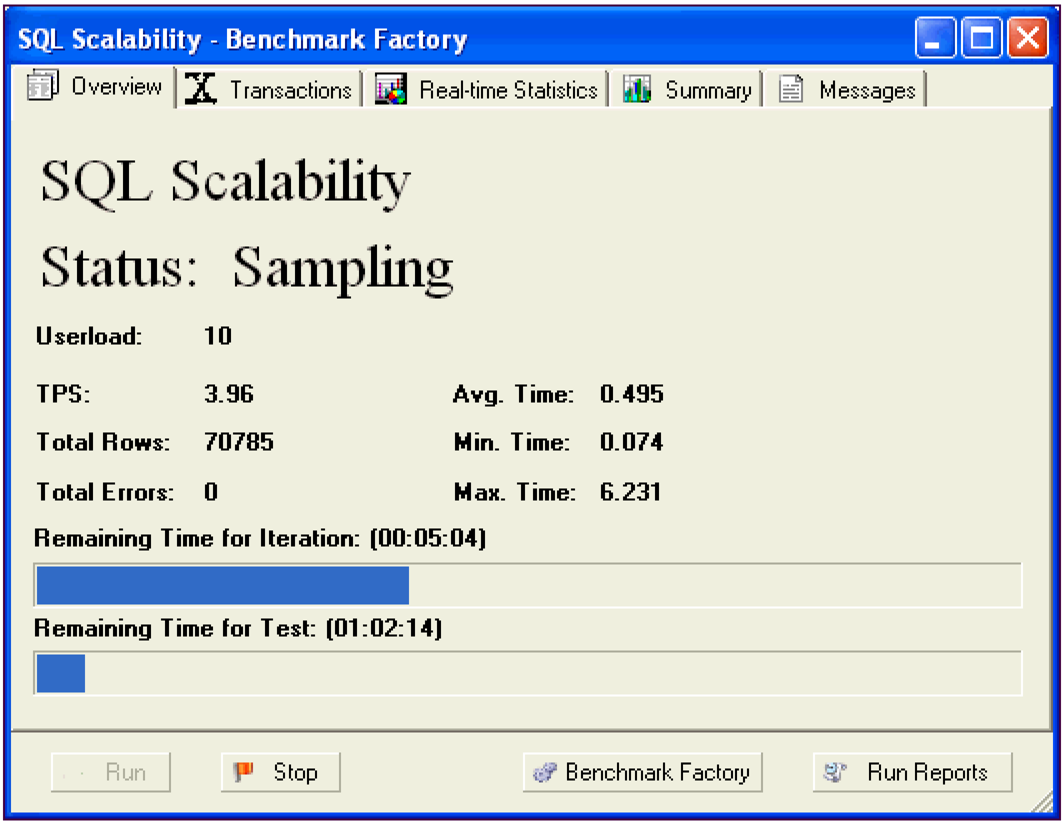
1. Click the Benchmark Factory drop-down icon and select the desired option. Three options are provided:

    2. **Test for Scalability**-Tests the currently displayed SQL.
    3. **Test All for Scalability**-Tests all SQL statements.
    4. **Test Selected for Scalability**-Tests the selected SQL statements.

2. The Benchmark Factory SQL Scalability dialog displays. Click **Next**. The Measurement Interval dialog displays.
3. Enter the desired user load.
4. Click **Next**. The Iteration Length dialog displays.
5. Enter the desired number of executions per iterations or executions per iteration.
6. Click **Next**. The Real World Latencies dialog displays.
7. Select the desired latency.
8. Click **Next**. The connection information dialog displays.
9. Enter the connection parameters.
10. Click **Next**. The Connection Agent dialog displays. Benchmark Factory Agent(s) reside on each physical client machine and spawn multiple virtual-user sessions. Each Agent generates an information screen with tabs that contain a variety of Agent information. Benchmark Factory can control hundreds of Agent machines.
11. If there are more than 20 users in your test, you will need to run an agent. Click **Start Local Agent**.
12. Click **Next**. The Benchmark Factory SQL test completion dialog displays.
13. Click **Finish**. The SQL Scalability dialog displays. From this dialog, you can control, monitoring, and view SQL testing results. This dialog provides the following tabs and buttons:

Overview Tab

The Overview tabs provides transactions per second testing results for individual user loads and iterations.



SQL Scalability

Status: Sampling

Userload: 10

TPS: 3.96 Avg. Time: 0.495

Total Rows: 70785 Min. Time: 0.074

Total Errors: 0 Max. Time: 6.231

Remaining Time for Iteration: (00:05:04)

Remaining Time for Test: (01:02:14)

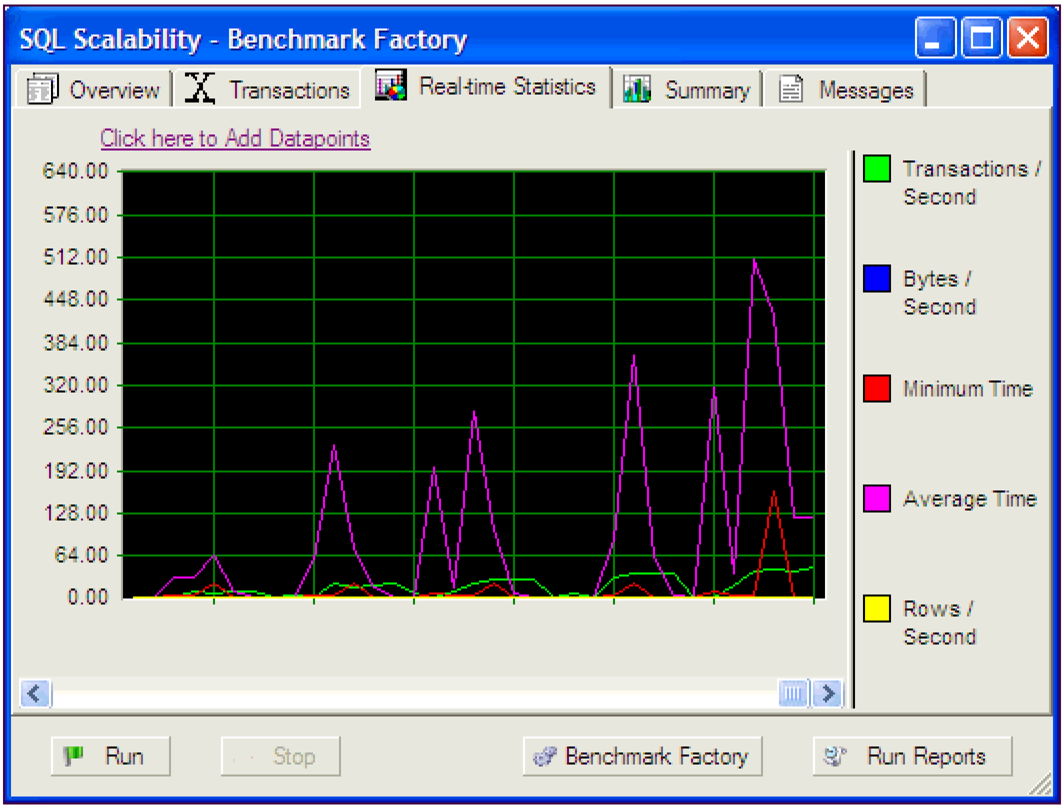
Transactions

The Transactions view provides the following statistics on the SQL statements being run:

- Executions
- Rows
- Bytes
- Errors
- Average Transaction Time
- Minimum Transaction Time
- Maximum Transaction Time

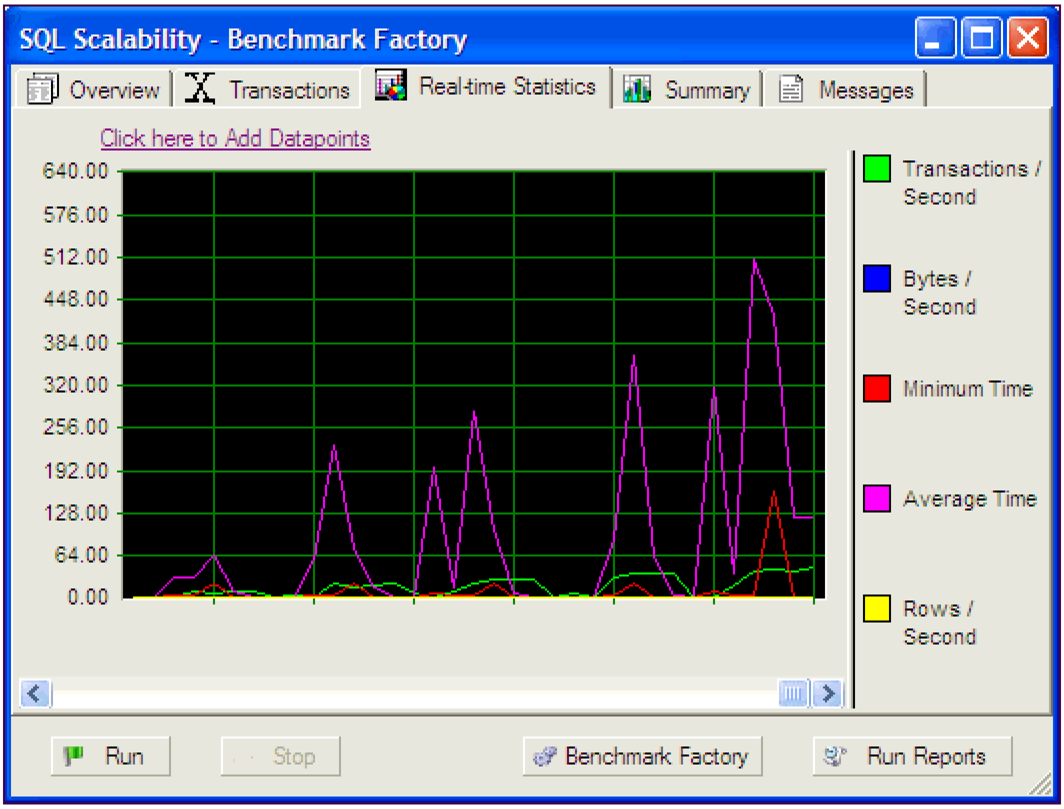
Real-Time

Real-Time Statistics provides real-time graphs and raw data. This data allows you to spot system-under test issues that may be affecting server performance. Right-clicking inside the graph displays a drop-down that allows you to change graph settings and view.



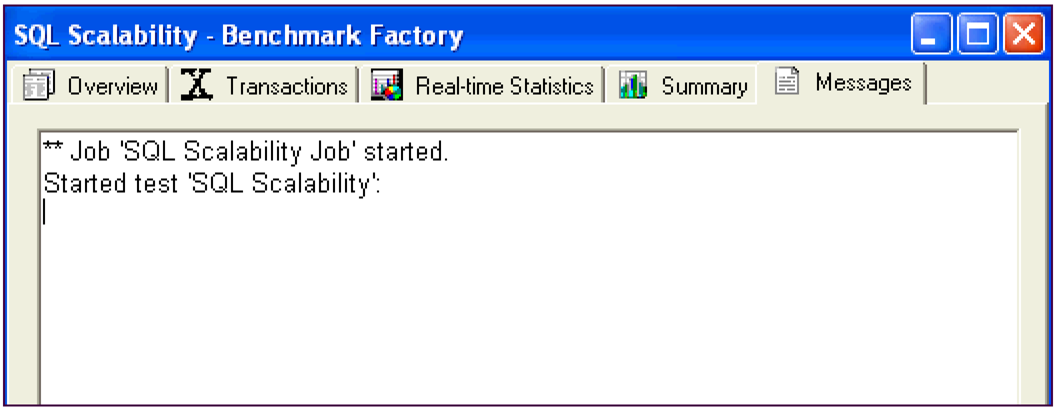
Summary

The Summary graph provides real-time and raw data summary graphs. The Summary graph displays real-time counter data. Right-clicking inside the graph displays a drop-down that allows you to change graph settings and view.



Messages

The Output window displays messages about the job being run.



```
** Job 'SQL Scalability Job' started.
Started test 'SQL Scalability':
```

Test Database Scalability

To test SQL Scalability, select one of the following:

- Create SQL Scalability Test
- Test Stored Procedures for Scalability

To test a system using standard benchmark tests, see the following:

- Create Industry Standard Benchmark Test

11

Settings

About Settings

Use the Settings dialog to specify or view the default settings Benchmark Factory uses when you create a new job. Changes to these settings affect only new jobs, not existing jobs.

To specify Benchmark Factory settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select a category from the list in the left pane. You can specify settings for the following:
   - General Settings
   - Benchmarks Settings - General
   - Benchmark Settings - Specific Test Types
   - Timing Settings
   - User Load Settings
   - Latency Settings
   - Error Handling Settings
   - Repository Settings
   - Statistics Settings
   - Agent Settings
   - Oracle Settings
   - SQL Server Settings
   - Execute File Settings

General Settings

Use the General tab of the Settings dialog to specify workplace settings. In addition, you can define the location for error logs and scripts.

To specify General settings

1. Select **Edit | Settings** (F8). The Settings dialog opens.
2. Select **General**. Review the following for additional information:

Group	Description
Directories	
Scripts	The default location used when saving scripts. These files usually reside on a local drive, but may reside on a network file server.
Error Logs	Location of error log files.
Data Files	Location of data files.
Result Logs	Location of result log files.
Retrieval Options	**Limit the number of rows retrieved when executing custom SQL transactions**—Allows you to limit the number of rows retrieved during SQL transaction execution.
Logs Cleanup	**Purge Error and Result log directories of files older than *N* days**—Specify the maximum number of days to retain log files. Files older than the time specified are deleted from the Error Logs and Result Logs directories when you click **OK** or **Apply**. Default = *180* **Purge Now**—Click to immediately delete log files older than the time specified. **Note:** .dmp files are not deleted.
TCP/IP Settings	**Console TCP/IP Port**—Specify a port for the Benchmark Factory console if different than the default. The default setting is port 4568. **Note:** Restart Benchmark Factory to apply changes.

3. Click **Apply** to apply changes.

   Note: You may need to restart Benchmark Factory to fully implement any changes to directories.

Benchmarks Settings - General

Use the Benchmarks page of the Settings dialog to specify some general benchmark testing options.

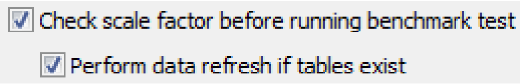
To specify general Benchmarks settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select **Benchmarks**. Review the following for additional information:

Refresh statistics after benchmark load	Select to instruct Benchmark Factory to refresh statistics after loading benchmark data. *Default = selected*
Check scale factor before running benchmark test	Instruct Benchmark Factory to check the Benchmark Scale factor of the existing tables against the new Benchmark Scale requirement before executing a test. To skip the scale checking process, do not select this checkbox. *Default = selected*

3. Select a default setting for the Object/Data Retention method. The default setting applies to new Create

Objects steps. Review the following for additional information:

Create/Load objects if objects don't exist (no backup sets created)	If objects do not exist, Benchmark Factory creates the objects and loads data. If you select this option, Benchmark Factory does not create backup tables.
Delete benchmark objects after completion of job	Benchmark Factory deletes objects after a job is finished.
Refresh data using backup sets	When backup sets exist, Benchmark Factory reloads data and recreates indexes. If backup sets do not exist, Benchmark Factory creates objects and loads data. Then Benchmark Factory creates the backup tables.
Recreate objects and reload data every execution	Benchmark Factory always deletes the existing objects and then recreates/reloads for each job execution. This allows you to ensure that the dataset is always in initialized status.
Refresh data using inserts	If data does not exist, Benchmark Factory uses bulk insert to load data. If the following Benchmark Settings are selected,  ☑ Check scale factor before running benchmark test ☑ Perform data refresh if tables exist and the Benchmark Scale factor of the existing tables does not match the new requirement, the process of inserting additional data or deleting data my run long, depending on the size of the existing data and number of indexes. **Note:** If database does not support bulk insert, insert is used.

4. Click **Apply** to apply your changes.

Benchmark Settings - Specific Test Types

For each standard benchmark test type, you can specify some default settings, such as scale factor, in the Settings dialog.

To specify default settings for a specific benchmark test type

1. Select **Edit | Settings | Benchmarks**.
2. Then select the ***benchmark test type***, for example TPC-C.
3. Specify default settings to be used when a new benchmark test is created. Review the following for additional information:

Database Size	Specify a database size. The Benchmark Scale readjusts according to the database size you specify.
Benchmark Scale	Specify a Benchmark Scale factor to be used to scale up table sizes and increase data. See Benchmark Scale Factor on page 134 for more

	information. The database size readjusts according to the scale factor you specify. After adjusting benchmark scale (or database size) review the estimates for individual and total table sizes in the Object Details grid.
Show Empty Tables	Select to display any tables that will be created but not populated with data.

4. Click **Apply** to apply your changes.

Note: The database and object sizes displayed are estimates. Actual sizes might vary based on database type and configuration.

TPC-B Benchmark Settings

Use this page of the Settings dialog to specify default values for creating TPC-B benchmark objects.

To specify TPC-B settings

1. Select **Edit | Settings | Benchmarks | TPC-B obsolete**.
2. Specify values to be used as the default settings when you create a new TPC-B Industry Standard Benchmark Test. Review the following for additional information:

Database Size	Specify a database size. The Benchmark Scale readjusts according to the database size you specify.
Benchmark Scale	Specify a Benchmark Scale factor to be used to scale up table sizes and increase data. See Benchmark Scale Factor on page 134 for more information. The database size readjusts according to the scale factor you specify. After adjusting benchmark scale (or database size) review the estimates for individual and total table sizes in the Object Details grid.
Show Empty Tables	Select to display any tables that will be created but not populated with data.

TPC-B History Tables

History tables allows you to set the number of history tables to create. The TPC-B benchmark is made up of only one transaction that updates three tables and inserts a record into a history table. Inserting one record into one history table limits testing performance.

This page of the Settings dialog allows the user to set a default value for the number of history tables to create during a test. The best ratio of history tables to virtual users is based on database configuration and hardware. The number of history tables to use is determined by the tester.

To specify the number of History tables to create

1. Select **Edit | Settings | Benchmarks | TPC-B obsolete | History Tables**.
2. In the **Number of History Tables** field, specify the number of tables you want to create by default.

Replication Benchmark Settings

Use the Replication page of the Settings dialog to specify default values for some options related to creating objects for the Replication benchmark test.

The default settings apply to new Create Objects for Replication steps.

To specify default Replication test settings

1. Select **Edit | Settings | Benchmarks | Replication**.
2. Specify values to be used as default settings when creating objects for a Replication benchmark test. Review the following for additional information:

Database Size	Specify a database size. The Benchmark Scale readjusts according to the database size you specify.
Benchmark Scale	Specify a Benchmark Scale factor to be used to scale up table sizes and increase data. See Benchmark Scale Factor on page 134 for more information. The database size readjusts according to the scale factor you specify. After adjusting benchmark scale (or database size) review the estimates for individual and total table sizes in the Object Details grid.
Show Empty Tables	Select to display any tables that will be created but not populated with data.

3. Click **Apply** to apply your changes.

Table Structure Settings

Use the Table Structure page of the Settings dialog to specify default values for the Replication tables.

The default settings apply to new Create Objects for Replication steps or new Replication tests.

To specify default table structure settings

1. Select **Edit | Settings | Benchmarks | Replication | Table Structure**.
2. Specify default settings to be used when creating objects for a Replication benchmark test. Review the following for additional information:

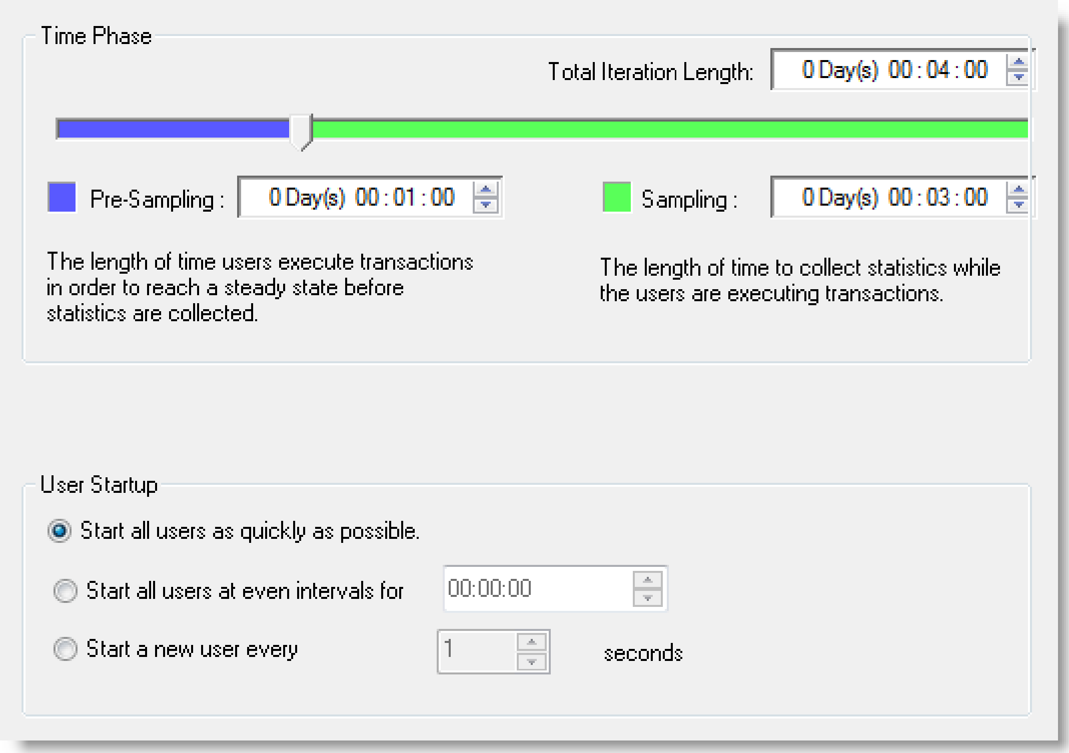
Number of Tables	Specify a default value for the number of tables to include in a new Create Objects for Replication step.
Number of Columns per Table	Specify a default range for the number of columns to include in each table in a new Create Objects for Replication step.
Add	Click to add a column.Then define the column details by entering values in the grid. See Replication Table Options Tab on page 97 for more information.
Ave. Number of Statements per Commit	Specify a default value for the number of statements to include in a commit.

3. Click **Apply** to apply your changes.

Timing Settings

The Timing page of the Settings dialog specifies timing phases associated with an item.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.



Time Phase

Total Iteration Length: 0 Day(s) 00 : 04 : 00

🟦 Pre-Sampling : 0 Day(s) 00 : 01 : 00

🟩 Sampling : 0 Day(s) 00 : 03 : 00

The length of time users execute transactions in order to reach a steady state before statistics are collected.

The length of time to collect statistics while the users are executing transactions.

User Startup

● Start all users as quickly as possible.

○ Start all users at even intervals for 00:00:00

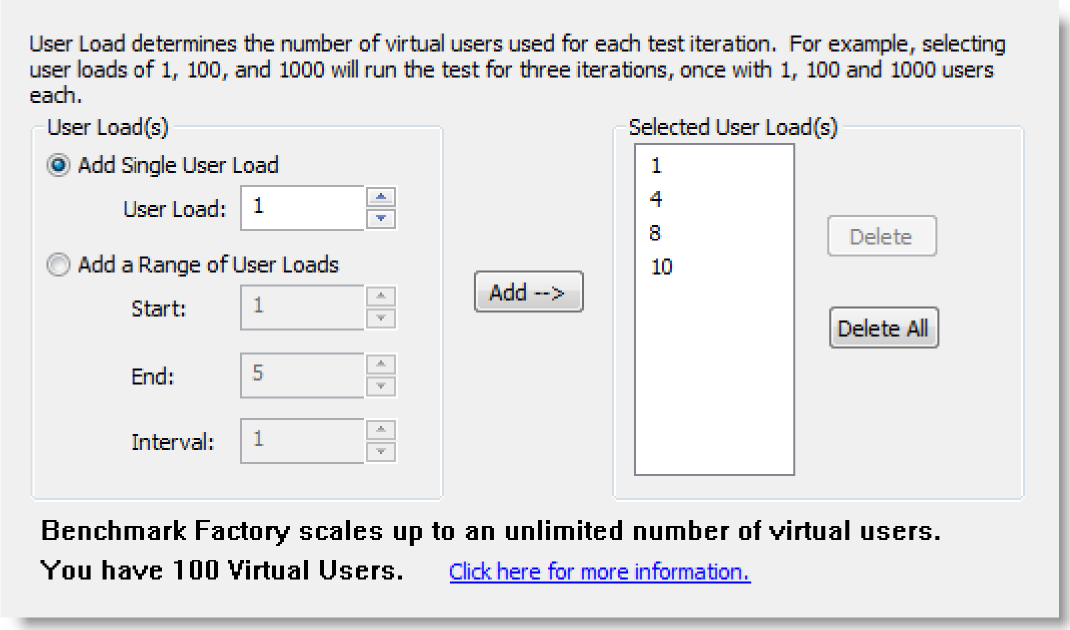
○ Start a new user every 1 seconds

Section	Field/Icon	Description
Time Phase	Total Iteration Length	The composite of the Pre-Sampling and Sampling time.
	Pre-Sampling	The length of time users execute transactions in order to reach a steady state before statistics are collected.
	Sampling	The length of time used to collect statistics while the users are executing transactions.
User Startup	Start all users as quickly as possible	Starts all users immediately after a test begins.
	Start all users at even intervals for	Sets the amount of time in which to start all users at even intervals. The interval duration is equal to this value divided by the number of users.
	Start a new user every [n] seconds	Starts a new user, then waits the entered number of seconds before starting the next user.

User Load Settings

You can use the User Load page of the Settings dialog to specify default settings for the number of virtual users for each iteration of a test. Then, when creating a new job in the New Job wizard, you can use these default settings or specify new settings for the job.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.



User Load determines the number of virtual users used for each test iteration. For example, selecting user loads of 1, 100, and 1000 will run the test for three iterations, once with 1, 100 and 1000 users each.

User Load(s)

● Add Single User Load

User Load: 1

○ Add a Range of User Loads

Start: 1

End: 5

Interval: 1

Selected User Load(s)

1
4
8
10

Benchmark Factory scales up to an unlimited number of virtual users.

You have 100 Virtual Users. Click here for more information.

Section	Field/Icon	Description
User Load	Add Single User Load	Specify a single user load, then click **Add** to add it to the selected user loads.
	Add a Range of User Loads	Specify a range, then click **Add** to add the range to the selected user loads.

Virtual Users

Benchmark Factory comes with 100 virtual users by default. See Add Virtual Users on page 232 for more information about adding virtual users.

Latency Settings

You can use the Latency page of the Settings dialog to specify default values for latency. Latency is delay added to the execution of a transaction to control of how fast transactions are submitted to the system-under-test. Use Latency to model real-world user interactions.

The default latency settings apply to transactions you add through the Transactions tab.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.

To specify default settings for Latency

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select the **Latency** page.
3. To specify no latency, select **No Delay**. To add latency, select one of the other options. Review the following for additional information:

No Delay	Select to specify no latency.
Transaction Rate	Select to set a transaction rate by specifying the interval between successive transactions arriving at the server. **Interarrival Time**—Select one of the following: • Absolute—Select to specify a fixed interarrival time. Then enter a value. • Distribution models—To use a distribution model for determining interarrival time, select a model and enter the requested value.
Real-World Delays	Use this option to simulate real-world processing by inserting a delay before and/or after each transaction execution. For each delay type, you can insert a fixed time by selecting *Absolute*, or you can select one of the distribution models for specifying time. **Keying Time**—Inserts a delay before execution to simulate the amount of time spent performing data entry. **Think Time**—Inserts a delay after execution to simulate the amount of time "to think" about the results of the previous transaction.

4. Select **Warning on interarrival time overrun** to display a warning message if a transaction runs longer than the interarrival time. The warning message is displayed in the Output window for the agent.

Error Handling Settings

Use this page to specify error handling options.

To specify error handling settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select **Error Handling**. Review the following.

Group	Description
Error Handling	
Stop test after first error	Default setting for a new jobs. Terminates the test when a server error occurs.
Stop job after first test error	Terminates the job when a server error occurs.
Thresholds	
Stop job if Memory usage exceeds	Instructs Benchmark Factory to stop the job if memory usage exceeds this threshold.

3. Click **Apply** to apply changes.

Repository Settings

ℹ **Note**: If you create a new repository in Benchmark Factory 5.5 or later, earlier versions of Benchmark Factory will not work against this repository.

The repository is a database where all of the test results are stored. Benchmark Factory inserts test results into the repository and provides an easy way to access all test results data.

By default, the Repository is a SQLite database that resides on the same machine as Benchmark Factory. The Repository can reside on another database server if required. To change the database, select the Data Source Name of the ODBC connection for the new database. To migrate data from one database to another, click **Data Migration** to open the Data Migration Wizard.

ℹ **Note:** By default in Benchmark Factory 7.1.1 or earlier, a MySQL database is created and used as the Repository, unless you selected the SQLite option during installation. In Benchmark Factory 7.2 or later, by default a SQLite database is created and used as the Repository.

If you plan to store a large amount of test data in the repository, you might want to consider using a more robust database than SQLite.

The Repository Settings page allows you to edit the DSN, perform ODBC administration, and test the connection. Benchmark Factory also provides a Repository Manager and Data Repository Migration wizard to assist you with other repository management functions.

ℹ **Note**: If the database structure does not exist on the selected database, Benchmark Factory prompts you to create the structure.

The supported database types are:

- MySQL
- SQL Server®
- SQLite
- SAP SQL Anywhere® (Adaptive Server® Anywhere)
- Oracle®

To specify Repository settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select **Repository**. Review the following for additional information:

Data Source Name	Data Source name of the ODBC connection used to connect to the repository database.
User Name	The User Name used to log into the selected database.
Password	The Password associated with the user name used to log into the database.
Edit DSN	Click to edit the selected Data Source Name.
ODBC Administrator	Click to open the ODBC Data Source Administrator dialog. Use this to add a DSN or ODBC connection.
Test Connection	Tests the defined connection.
Data Migration	Click to open the Data Migration Wizard. See Data Repository Migration Wizard on page 227 for more information.

Statistics Settings

You can use this page of the Settings dialog to specify default values for the statistics collection options.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.

To specify Statistics settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Specify statistics collection options. Review the following for additional information:

Save results to Repository	Select to save test information to the Repository. Then select one or more of the following options: • **Save Real-Time Counter Information**—Saves real-time performance monitoring information to the Repository during the test. See Performance Counters Tab on page 111 for more information. **Note:** To specify sampling rate, go to **Edit | Settings | Statistics | Real-Time counters**. • **Save User Statistics**—Saves user-level statistics. • **Save Distribution**—Saves transactions per second and response time per user distribution. • **Log Transaction Results to disk**—Logs the transaction results of a job to disk. **Note:** Checking this option imposes performance penalties.

To specify Real-Time Counter settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Expand the Statistics node and select **Real-Time Counters**.
3. In the **Real-time reporting interval** field, specify a sampling rate.

Tip: You can over-ride the default settings and specify statistics collection options per job. In the New/Edit Job Wizard, select a test under **Workload** and then select the Performance Counters tab (under Job Setup tab).

Agent Settings

You can use this page of the Settings dialog to:

- Specify some default settings (error handling and communication timeout) for each agent
- Set up (add) new agents to use in testing
- Specify the defaults agents to use when creating a new job.
- Install remote agents

To specify default Agent settings

1. Select **Edit | Settings**. The Settings dialog opens.
2. Select **Agent**. Review the following for additional information:

Setup New User Agent	Click to setup a new agent.
List of agents	Specify default settings. Select the agent machines and specify the number of agents. These default settings are used when you create a new job. If you do not select a default agent, the local agent is automatically selected when you create a new job in the New Job wizard. **Note:** When you run a job using one or more local agents, if Agent utilization of resources on the local machine is too high, errors could occur.
Remove	Click to remove the selected agent.
Verify Selected Agent	Verifies that selected agent starts and communicates with console.
Error Handling	**Stop job when agent cannot be started/connected**—Cancels job if agent does not start or connection is lost. Do not select this option if you want to run a job using multiple agents AND you want the job to run even if one or more agents do not start.
Agent communication timeout	Specify timeout duration in seconds.
Agent refresh interval	Specify how often each agent should refresh its status.

To set up or to install an Agent

1. Select **Edit | Settings | Agent**.
2. On the Agent settings page, click **Setup New User Agent**. The Setup User Agent dialog opens.
3. Enter the name or IP address of the computer where the agent is installed, or the computer on which you want to install an agent.
    - To install an agent on a remote computer, see Install Remote Agents.
4. Enter the connection information.
5. Specify the number of agents. Click **OK**.
6. Benchmark Factory connects to the computer and then checks if the agent is installed.
7. The agent is added to the list.
    - To learn more about running multiple agents, see Running Benchmark Factory with Multiple Agents.

Tips:

- You can also modify agents or set up new agents for a job using the New/Edit Job Wizard. Select **Agent** in the left pane of the wizard to access agent options for the selected job.
- To open the Agent console, go to **Program Files\Quest Software\Benchmark Factory <version>\bin** and double-click **Agent.exe**. See About Agents on page 36 for more information.

Oracle Settings

You can use the Oracle page of the Settings dialog to specify default settings for testing against an Oracle database. Use this page to select the performance collection and reporting methods and the data cache clearing options.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.

To specify Oracle settings

1. Select **Edit | Settings | Oracle**.
2. Review the following for additional information:

Statspack Options	
Perform Statspack snapshot during each iteration	Select to use Oracle’s “Stats Pack” utility to collect statistics. • Number of snapshots—Specify the number of snapshots.

AWR and ADDM Options	
Perform AWR snapshot at the start and end of each test iteration	Select to use the optional Oracle Enterprise Manager (OEM) “Diagnostic Pack” Automatic Workload Repository (AWR) to collect statistics. Then select the type of report to generate: • Generate AWR report • Generate ADDM report
Perform AWR snapshot during each iteration	Select to use Oracle AWR to collect statistics. • Number of snapshots—Specify the number of snapshots.

Note: A valid license is required to use the optional Oracle Enterprise Manager (OEM) Diagnostics Pack.

Database Flush	
Flush data buffer caches at start of each test iteration	Select to clear data buffer caches between iterations. **Note:** To perform this action, the Oracle database account must have certain privileges. In Oracle 9i, the ALTER SESSION privilege is required. In Oracle 10g or later, the ALTER SYSTEM privilege is required.
Flush shared pool at start of each test iteration	Select to clear shared pool between iterations. **Note:** To perform this action, the Oracle database account must have the ALTER SYSTEM privilege.

Note: Cached data can improve performance, so selecting one or both of these options can prevent cached data from affecting subsequent iterations.

Tip: You can specify these same Oracle settings for each individual Oracle connection. See Create Oracle Connection and Oracle Statistics Tab (Connections) for more information.

SQL Server Settings

Use the SQL Server page of the Settings dialog to specify default settings for testing against SQL Server databases.

Note: If you modify the default settings in the Settings dialog, the changes apply to new jobs only, not to existing jobs.

To specify SQL Server settings

1. Select **Edit | Settings | SQL Server**.
2. Review the following for additional information:

Database Flush	
Clean data buffer and procedure caches at start of each test iteration	Select this option to instruct Benchmark Factory to clear cached data between iterations. Cached data can improve performance, so selecting this option can prevent cached data from affecting subsequent iterations. **Notes:** • This option is only applicable to SQL Server 2005 or later. • To perform this action, the SQL Server database account must have the sysadmin fixed server role.

Tip: You can specify this option for each individual SQL Server connection. See Create SQL Server Connection on page 48 for more information.

Execute File Settings

Use the Execute File page of the Settings dialog to specify default settings for the Execute External File step/test.

To specify Execute File settings

1. Select **Edit | Settings | Execute File**.
2. **Enforce Timeout**—Specify a default setting. Selecting the option instructs Benchmark Factory to enforce a timeout for the Execute File step/test. Then enter the number of seconds for Benchmark Factory to wait for the step/test execution to complete before continuing the job.

   These default settings are used when you create a new job in the New Job wizard.

12

Test Results and Run Reports

Benchmark Factory Test Results

Analysis is the ultimate goal of your load test. When you run a test, you get results both in real-time and in a saved format. Benchmark Factory provides you with testing results that are easy to interpret and allows you to attribute individual results to individual tasks and users.

Graphs

Whether you are viewing a real-time or historical graph, Benchmark Factory uses the same graphing tool. In most cases you will find that the graphs are presented in the most meaningful form. The Benchmark Factory graphing tool allows you to customize graphs to tailor your load testing viewing requirements.

View Test Results

Benchmark Factory provides the following ways to view test results:

- Real-time statistics and progress—View job progress and real-time statistics while a job is running.
- After job completion (results summary)—View a summary of test results and compare results from different tests.
- After job completion (Run Reports)—View detailed test results in Run Reports.
- Export results to Excel—You can export test results to Excel from the Results/Compare Results page or Run Reports.
    - Exported report information to Excel using the command line

View Real-Time Test Results

View Test Results - Running Job

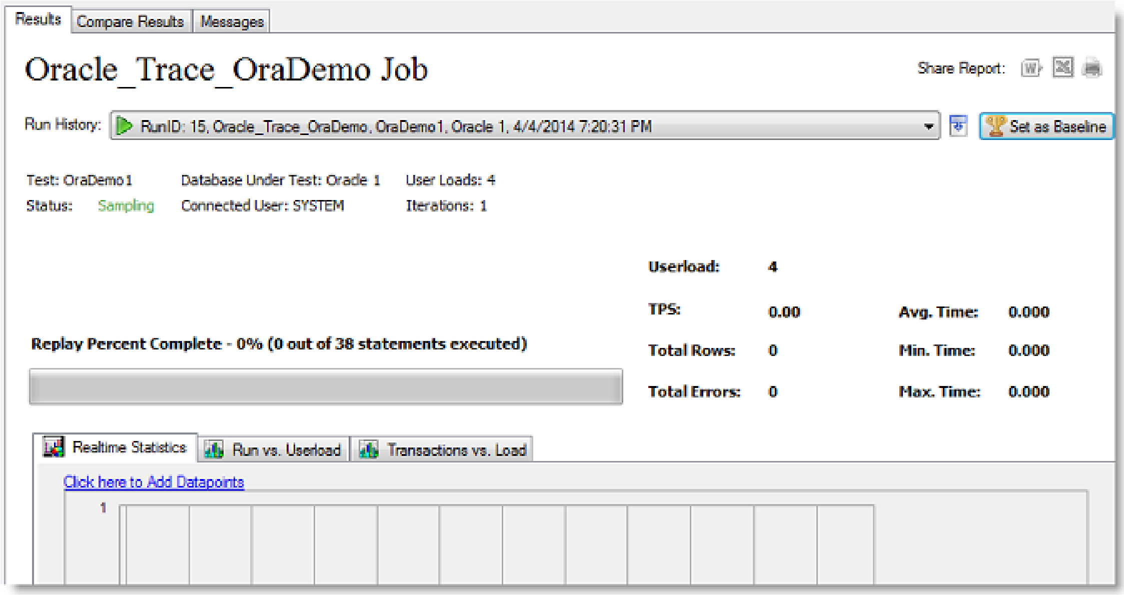
While a job is running, the Results page displays job progress and real-time statistics information. You can monitor the status of the test, see which user load its running, view transactions per second (TPS), and view a variety of other data points that give you insight into the actual performance your database provides under the user load being tested.

You can also view real-time statistics while a job is running. Real-time statistics provide insight into the performance of the database being tested. Numerous data points allow you to examine exactly how your database is performing.

See also, View Test Results - Completed Job.

To view progress and results during job execution

1. Select a running job in the Jobs View pane.
2. The Results page displays. This page shows the status of the job, the progress of the current step that is executing (such as loading data or sampling), as well as other data such as which user load is running.



Oracle_Trace_OraDemo Job

Run History: RunID: 15, Oracle_Trace_OraDemo, OraDemo1, Oracle 1, 4/4/2014 7:20:31 PM

Test: OraDemo1 Database Under Test: Oracle 1 User Loads: 4
Status: Sampling Connected User: SYSTEM Iterations: 1

Userload: 4
TPS: 0.00 Avg. Time: 0.000
Total Rows: 0 Min. Time: 0.000
Total Errors: 0 Max. Time: 0.000

Replay Percent Complete - 0% (0 out of 38 statements executed)

To view real-time statistics during job execution

1. Select a running job in the Jobs View pane.
2. The Results page displays. In the lower portion of the Results page, select one of the tabs to view real-time statistics. Review the following for more information:
   - Realtime Statistics Tab
   - Run vs Userload Tab
   - Transaction vs Load Tab

Iteration Overruns

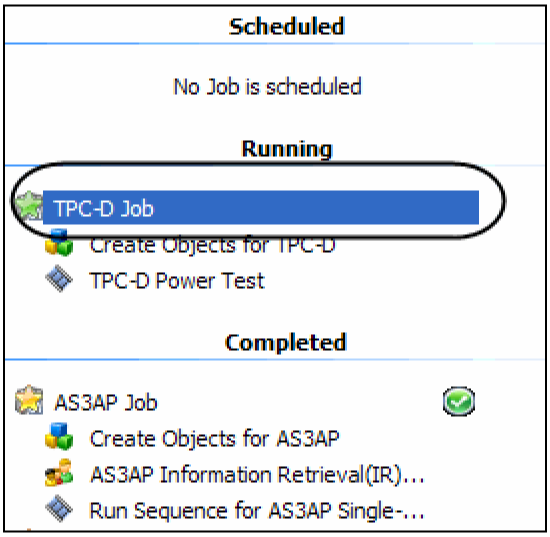
Iteration overruns occur at the end of an iteration to allow time for all transactions submitted within the test iteration cycle to complete, so that all transaction statistics can be collected. For example, an agent may execute a transaction during the last five seconds of test iteration, if this transaction takes 15 seconds to complete, an iteration overrun of 10 seconds will occur.

Realtime Statistics Tab

The Realtime Statistics tab in the Results page provides you with real-time graphs and raw data. This data allows you to spot system-under-test issues that may be affecting server performance. The Realtime Statistics tab displays real-time counter data. Right-clicking inside the graph displays a drop-down that allows you to change graph settings and view.

To view Real-time Statistics

1. In the Jobs View, click the job running.



Scheduled

No Job is scheduled

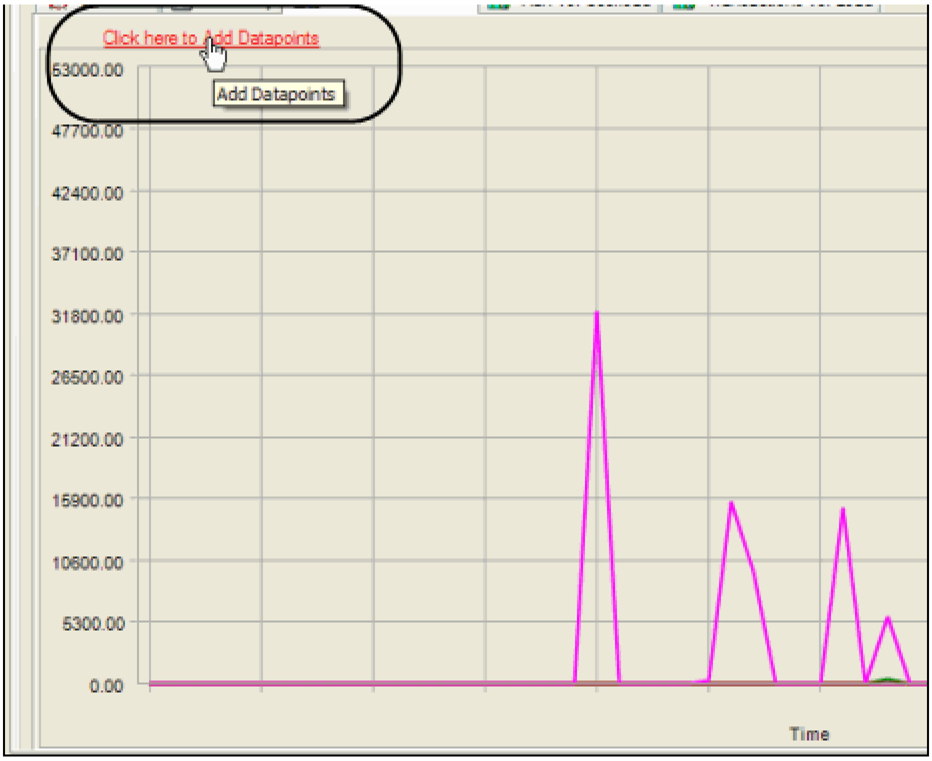
Running

- TPC-D Job
  - Create Objects for TPC-D
  - TPC-D Power Test

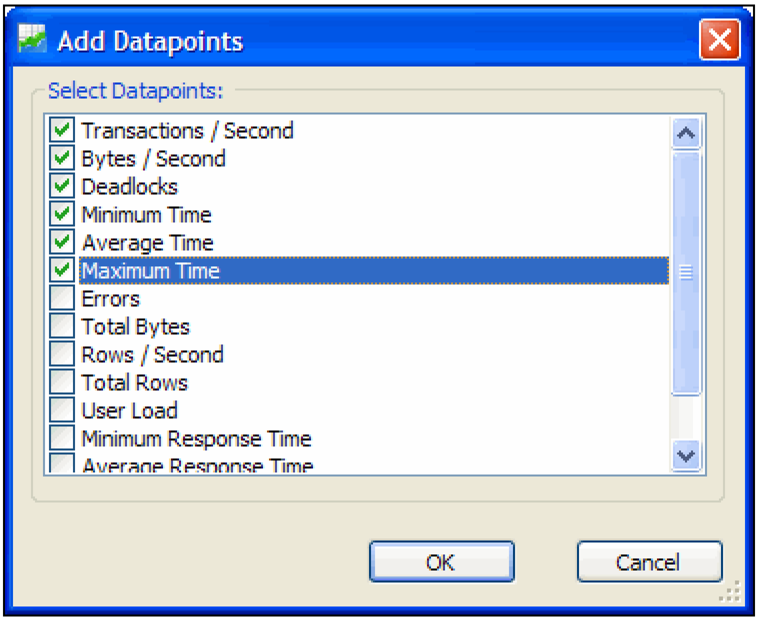
Completed

- AS3AP Job
  - Create Objects for AS3AP
  - AS3AP Information Retrieval(IR)...
  - Run Sequence for AS3AP Single-...

2. Select **Click here to Add Data points**. The Add Data points dialog displays.



3. Select the desired data points to view.

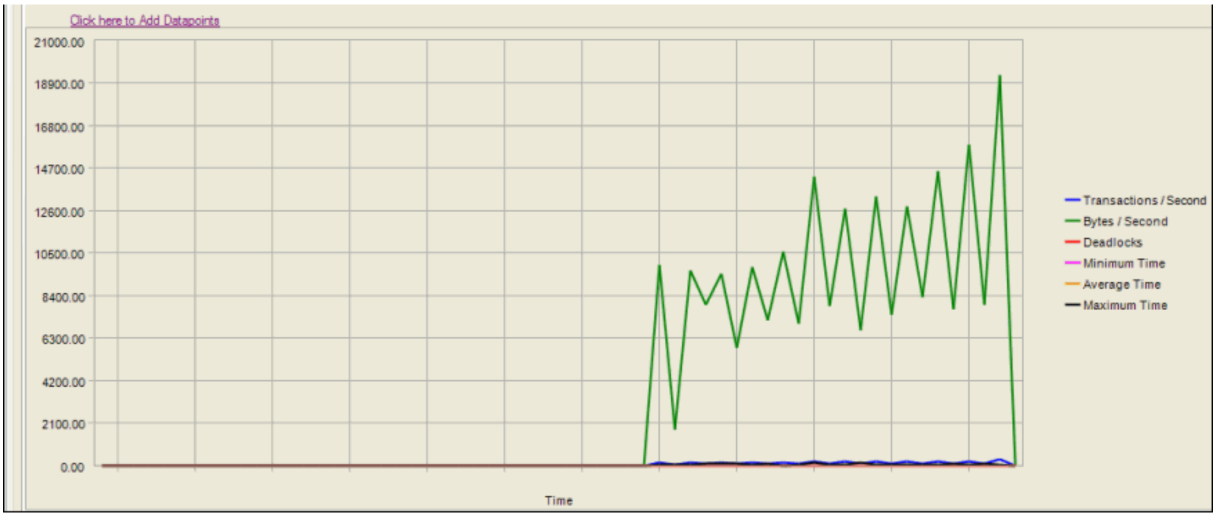


Add Datapoints

Select Datapoints:

- ☑ Transactions / Second
- ☑ Bytes / Second
- ☑ Deadlocks
- ☑ Minimum Time
- ☑ Average Time
- ☑ Maximum Time
- □ Errors
- □ Total Bytes
- □ Rows / Second
- □ Total Rows
- □ User Load
- □ Minimum Response Time
- □ Average Response Time

3. Click **OK**. The graph displays with the selected data points. (To change graph views, right-click.)

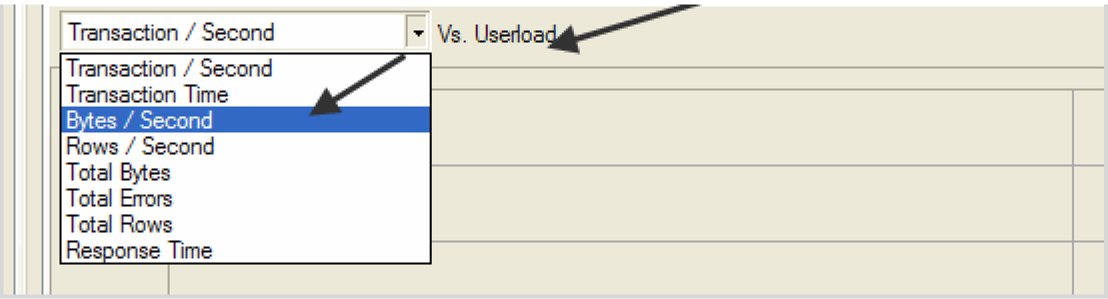


Run vs. Userload Tab

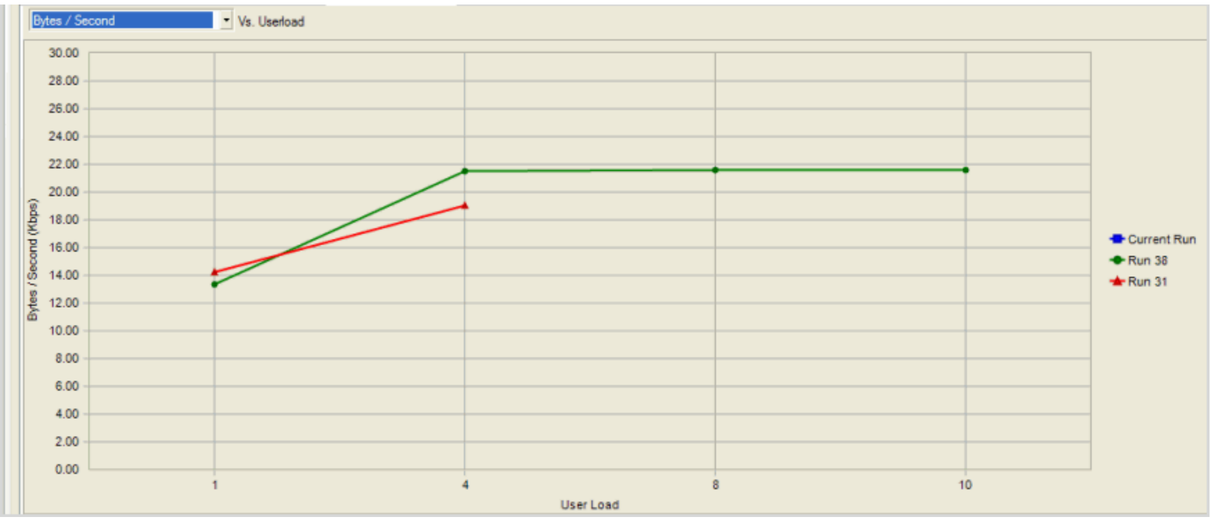
The Run vs. Userload tab in the Results page compares the user load transaction against the selected data point allowing you to view how the various data points are being affected by the job running.

To view Run Vs. Userload tab testing results:

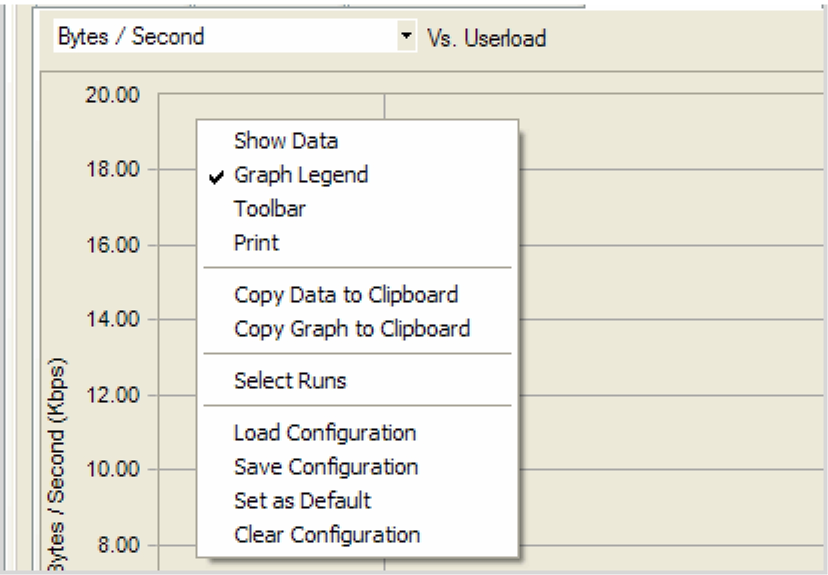
1. While the job is running, select the **Run vs. Userload** tab on the Results page.
2. Click the data points Vs. Userload drop-down and select the desired data point.



The graph updates with the selected data point.



3. To change graph views, right-click. See Change Graph Views for more information.

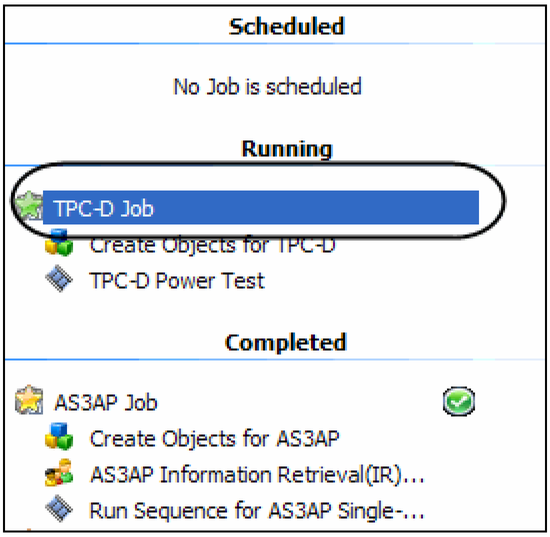


Transactions vs Load Tab

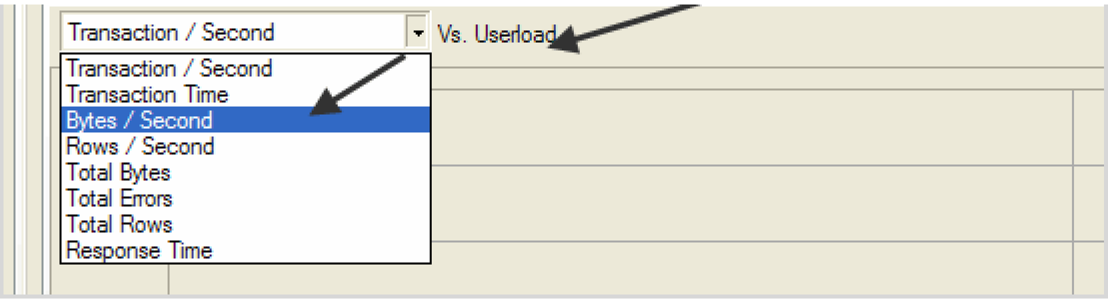
The Transactions vs Load tab in the Results page displays user selected transactions statistics for the job timing period following completion of an iteration.

To view the Real-time statistics

1. In the Jobs View, click the job running.



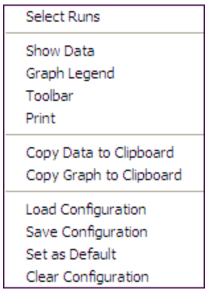
2. Click the *data points* Vs. Userload drop-down.



3. Select the desired data points to view.
4. To change graph views, right-click. A drop-down displays.
5. Select the desired option.

Change Graph Views

Right-clicking on a Benchmark Factory graph displays a drop-down that allows you to customize graph settings. Whether you are viewing a real-time graph or one in Benchmark Factory Run Reports, Benchmark Factory uses the same graphing tool. In most cases you will find that the graphs are presented in the most meaningful form (Line Graphs or Bar Graphs based on the data being presented.)



Displays Benchmark Factory Run Reports

Shows in a data form or graph form

Displays the Graph Legend

Displays the graph toolbar

Prints the graph

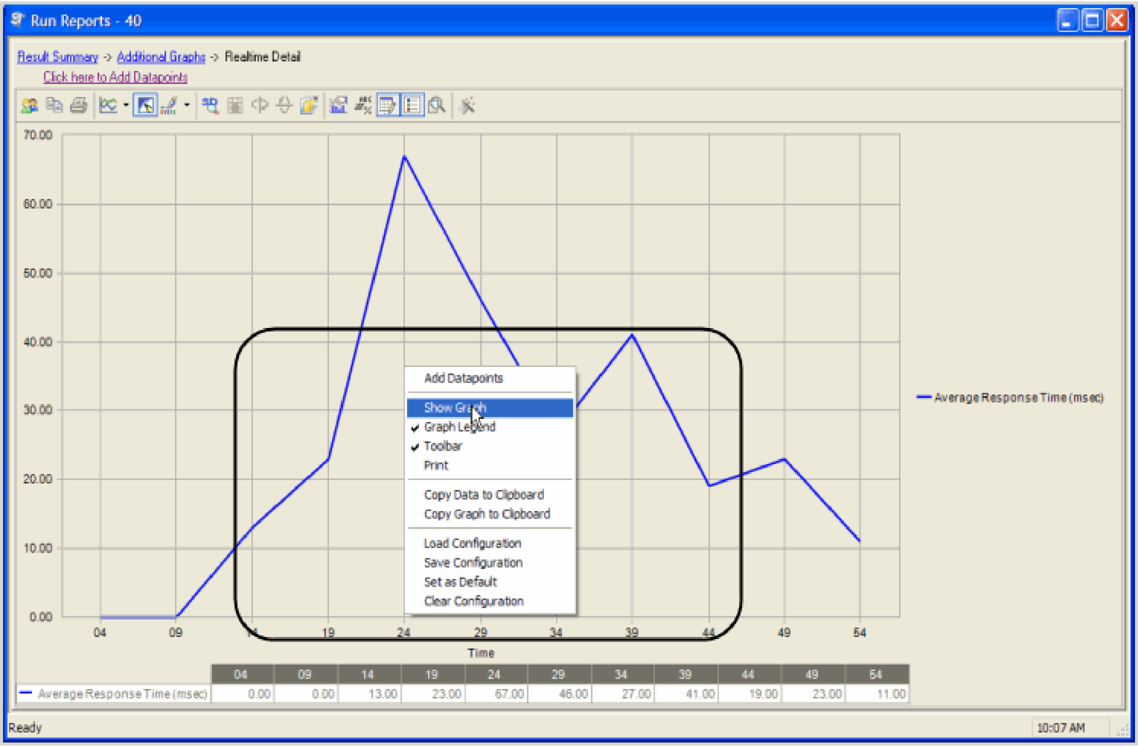
Copies Data to Clipboard

Copies Graph to Clipboard

	Loads a saved graph configuration Saves a graph configuration Sets the current configuration as default Clears the current chart configuration

Show Data/Show Graph

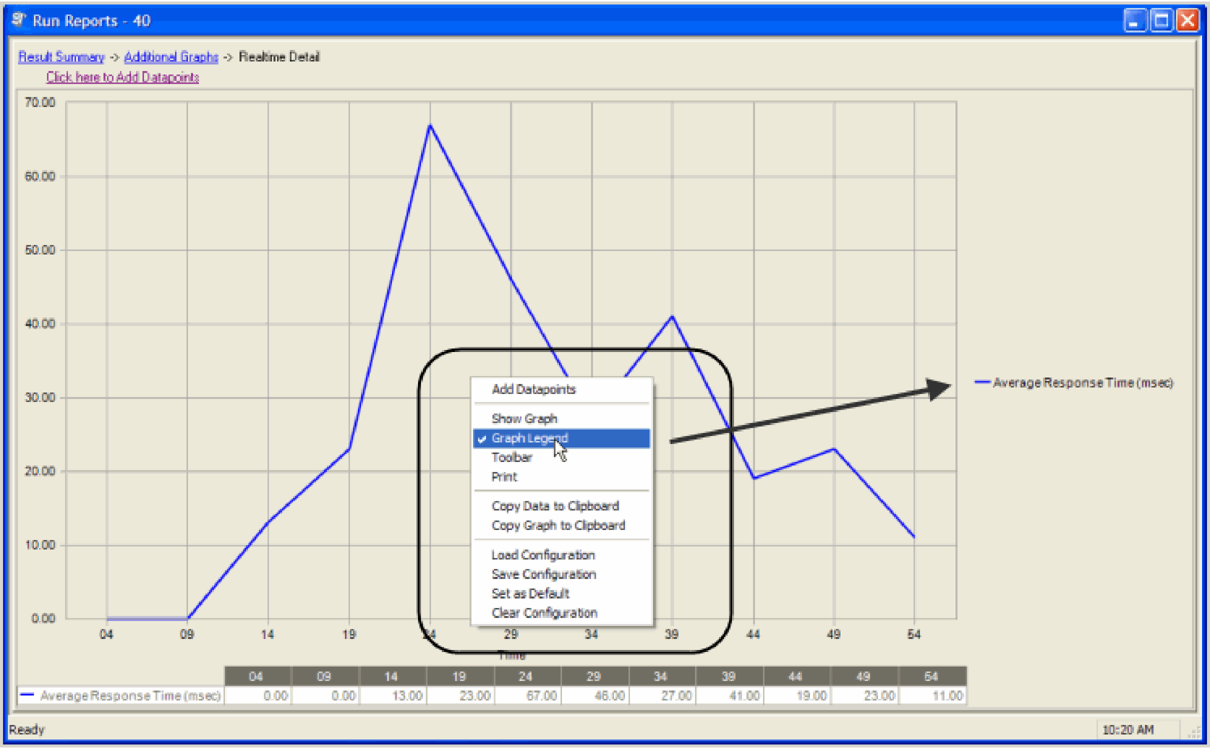
Toggling to Graph displays the data in graph form.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Graph Legend

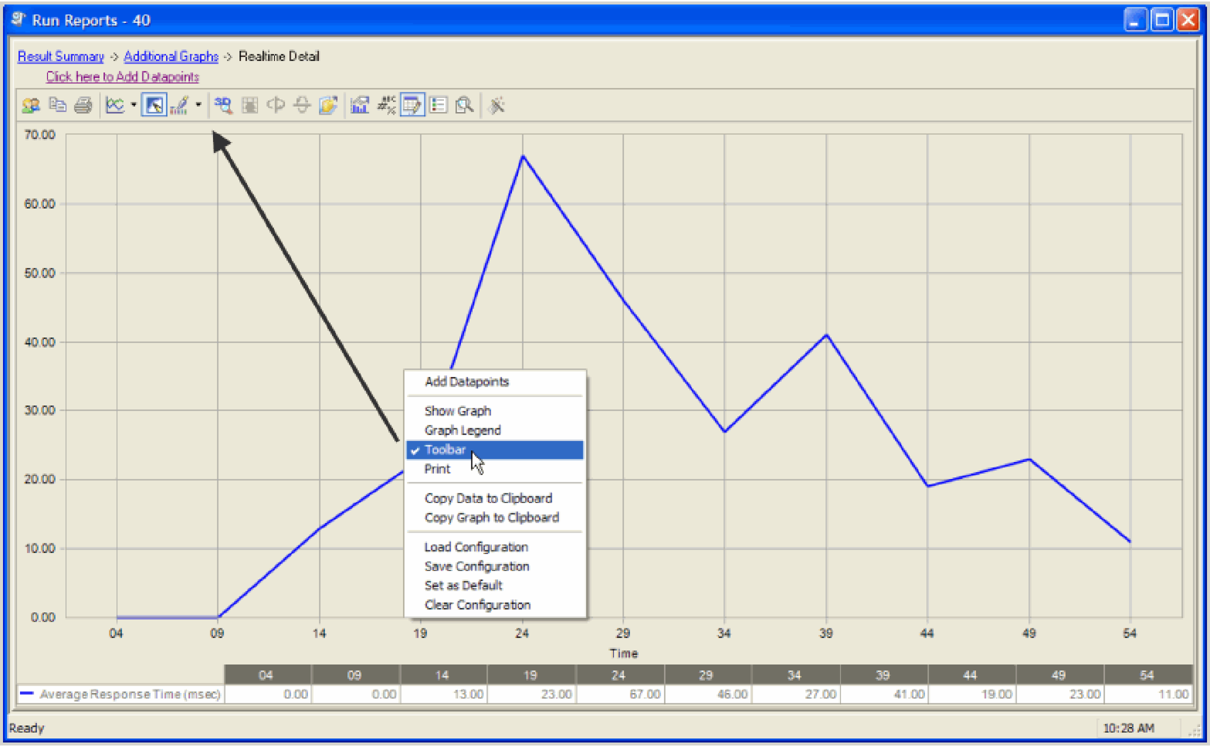
Toggling to Graph Legend displays a legend on the side of the graph.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Toolbar

Toggling to Toolbar displays the graph toolbar.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Print

Choosing Print displays the Print Dialog.

Copy Data to Clipboard

Copies the data to the clipboard.

Copy Graph to Clipboard

Copies the graph to the clipboard.

Load Configuration

Benchmark factories graphs allows you to save graph configurations.

Save Configuration

Saves a graph configuration.

Set as Default

Sets a configured graph as default.

Clear Configuration

Clears a graph configuration.

View Test Results - Completed Job

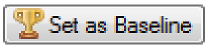
After a job is finished running, you can view a summary of test results in the Results page. Use the Compare Results page to compare different test runs.

See also, View Test Results - Running Job.

To view results after job completion

1. Select a job in the Jobs View pane.
2. The Results page displays.
3. Click ⇩ to show all test runs.
4. Select a run to review a summary of test results. Review the following for additional information:

W	Click to export test results to Word from one or more selected runs.
X	Click to export test results to Excel from one or more selected runs.
→	Click to export the selected run's test results as a zip file. The zip file contains an XML file and other files required to reproduce the test results.
↲	Click to import test results from a zip file exported from Benchmark Factory.

	Click to view one or more selected test run results in Run Reports. Run Reports provides more-detailed test results.
	Sets the selected run as the baseline. This allows you to easily compare other run results against this baseline. Click **Reset Baseline** to clear baseline selection.

Compare Results

You can use the Compare Results tab to compare test results between selected test runs, or to compare results to the baseline test run.

To compare results

1. Select a job in the Jobs View pane.
2. The Results page displays.
3. Select the **Compare Results** tab.
4. In the top grid, use Ctrl+Click to select the test runs to compare. The selected test runs display in the Run Comparison grid.

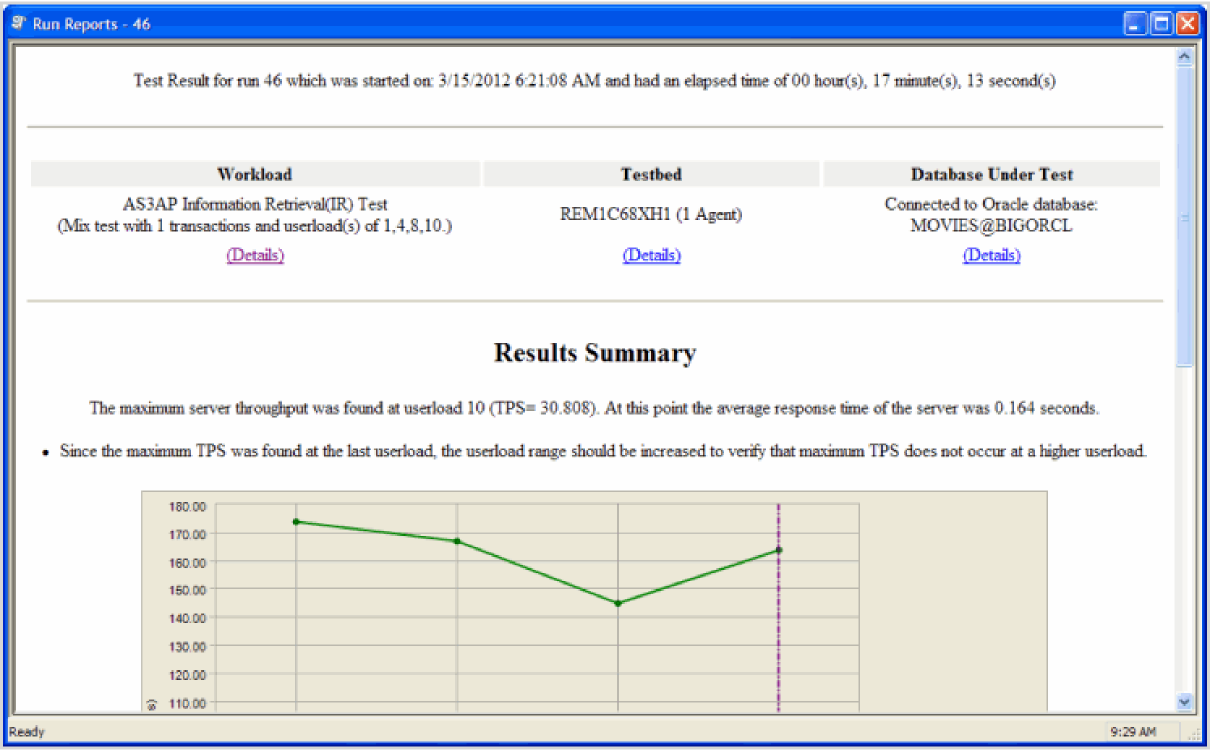
   You can use the links above the top grid to filter the list of test runs to display.
5. Select the **Automatically compare to baseline** checkbox to always compare the selected run to the baseline.

Benchmark Factory Run Reports

Run Reports

Benchmark Factory Run Reports is a separate executable that provides a comprehensive and detailed collection of database load testing results. With Benchmark Factory you can drill down into a database to view a wide array of information and statistics that gives you accurate insight into database performance. Run Reports Viewer allows you to access Benchmark Factory load testing results.

ℹ **Note**: Three instances of Run Reports can be viewed at one time.



Test Result for run 46 which was started on: 3/15/2012 6:21:08 AM and had an elapsed time of 00 hour(s), 17 minute(s), 13 second(s)

Workload	Testbed	Database Under Test
AS3AP Information Retrieval(IR) Test (Mix test with 1 transactions and userload(s) of 1,4,8,10.) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

Results Summary

The maximum server throughput was found at userload 10 (TPS= 30.808). At this point the average response time of the server was 0.164 seconds.

- Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

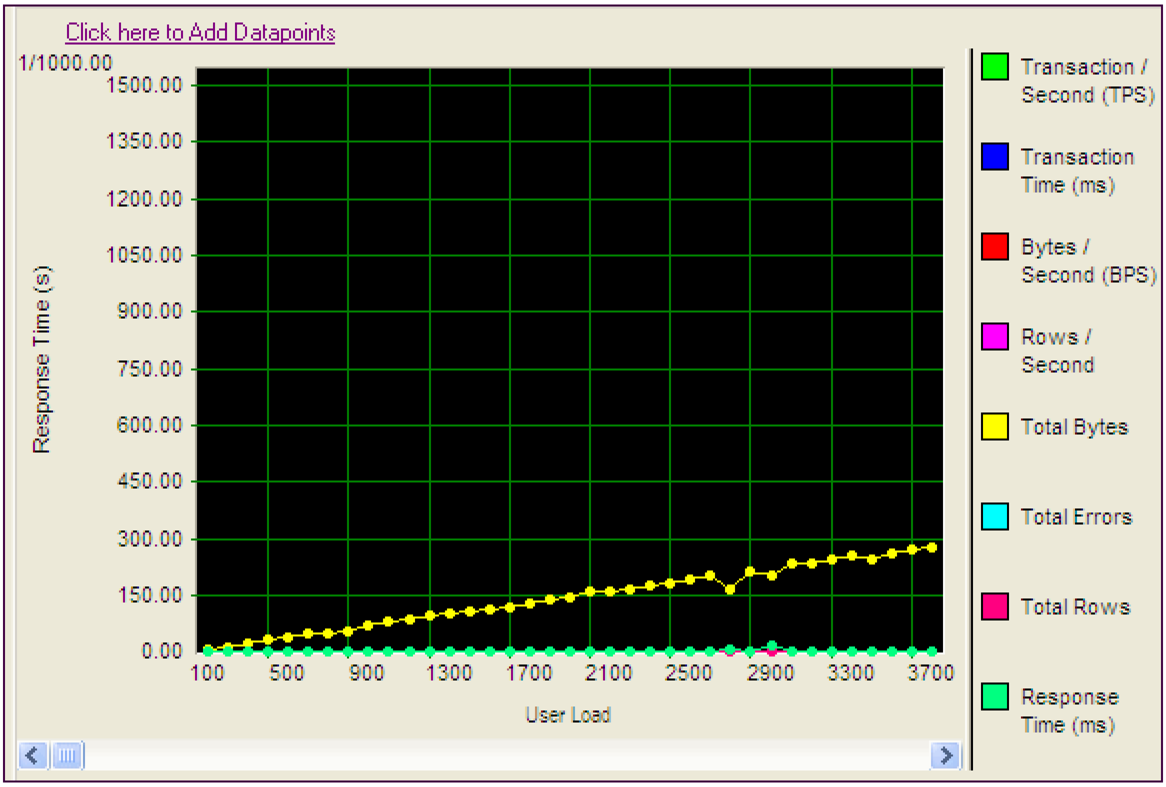
Cutting and Pasting of Run Reports

From Benchmark Factory run reports you can cut and paste test results into the Clipboard.

Testing Server Throughput

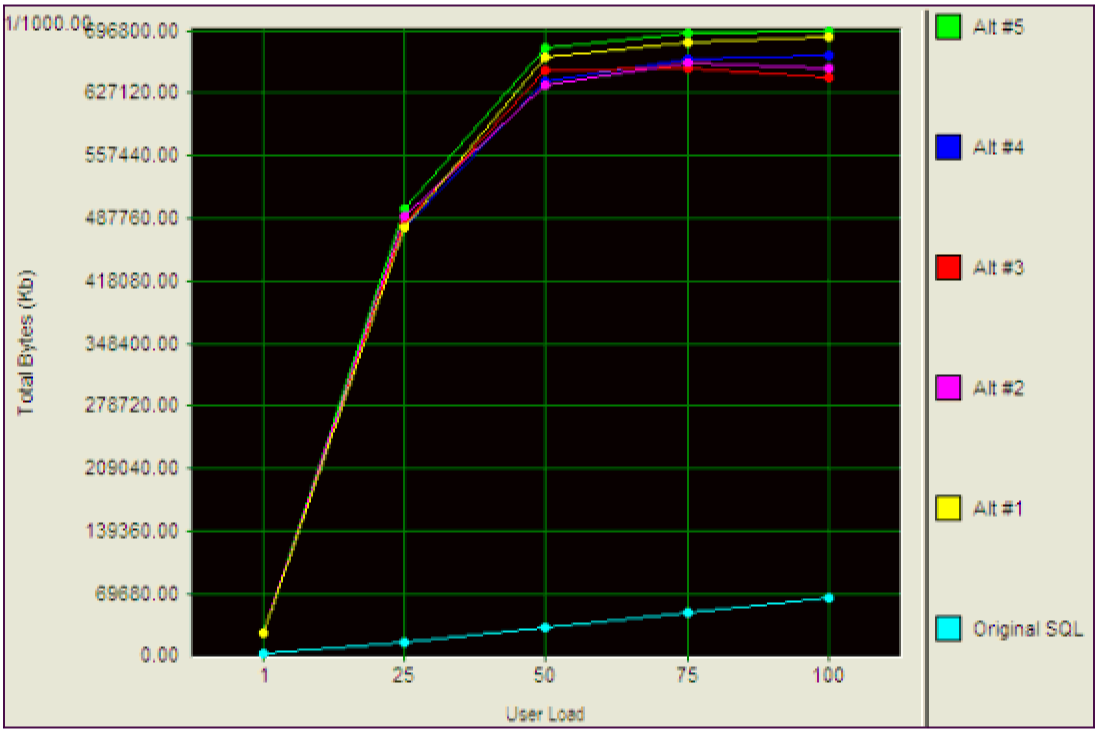
Server throughput is the measurement of database capacity. Benchmark Factory allows you to view a variety of server throughput statistics that include:

- Transaction/Second (TPS)
- Transaction Time
- Bytes/Second (BPS)
- Rows/Second (RPS)
- Total Bytes
- Total Errors
- Total Rows
- Response Time



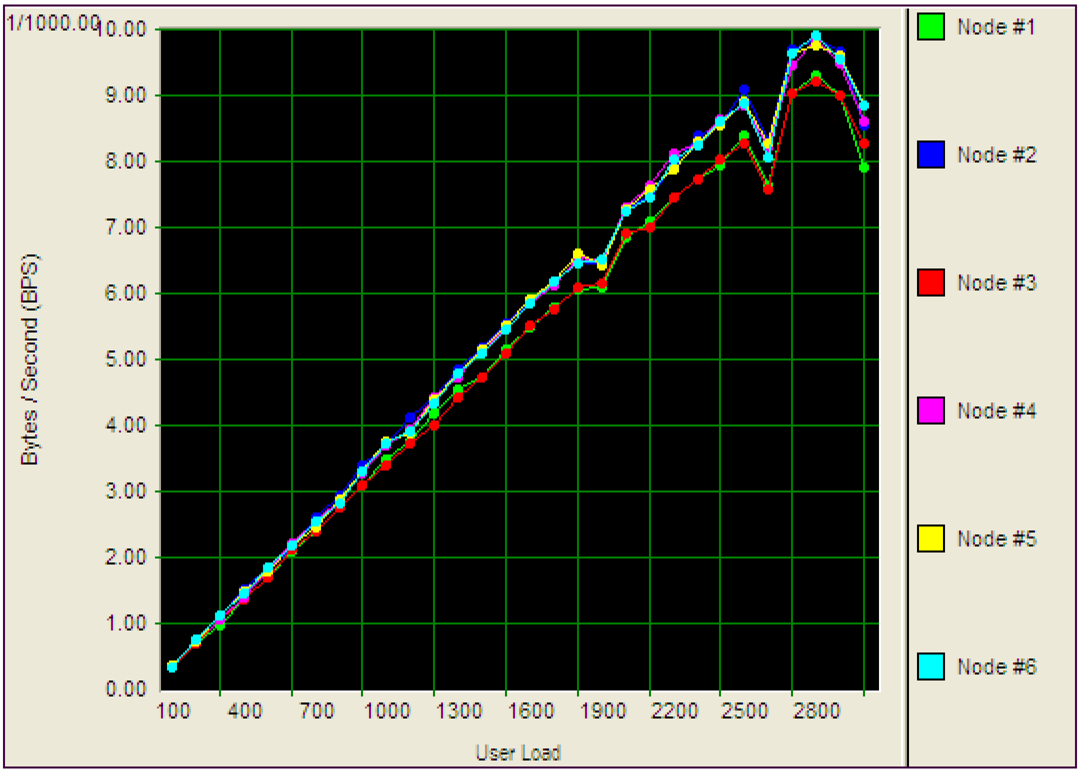
Testing SQL Scalability

The Benchmark Factory transaction comparison graph provides users with a comparison of the individual transaction performance as they compare to each other. This is most useful when analyzing a SQL Scalability test.



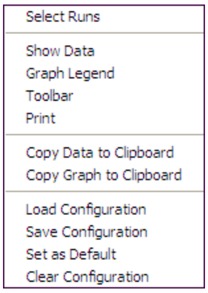
Testing Real Application Clusters

Benchmark Factory allows you to load test Oracle Real Application Clusters (RAC) to determine system capacity. The cluster performance graph allows users to view the performance of individual nodes.



Change Graph Views

Right-clicking on a Benchmark Factory graph displays a drop-down that allows you to customize graph settings. Whether you are viewing a real-time graph or one in Benchmark Factory Run Reports, Benchmark Factory uses the same graphing tool. In most cases you will find that the graphs are presented in the most meaningful form (Line Graphs or Bar Graphs based on the data being presented.)



Displays Benchmark Factory Run Reports

Shows in a data form or graph form

Displays the Graph Legend

Displays the graph toolbar

Prints the graph

Copies Data to Clipboard

Copies Graph to Clipboard

Loads a saved graph configuration

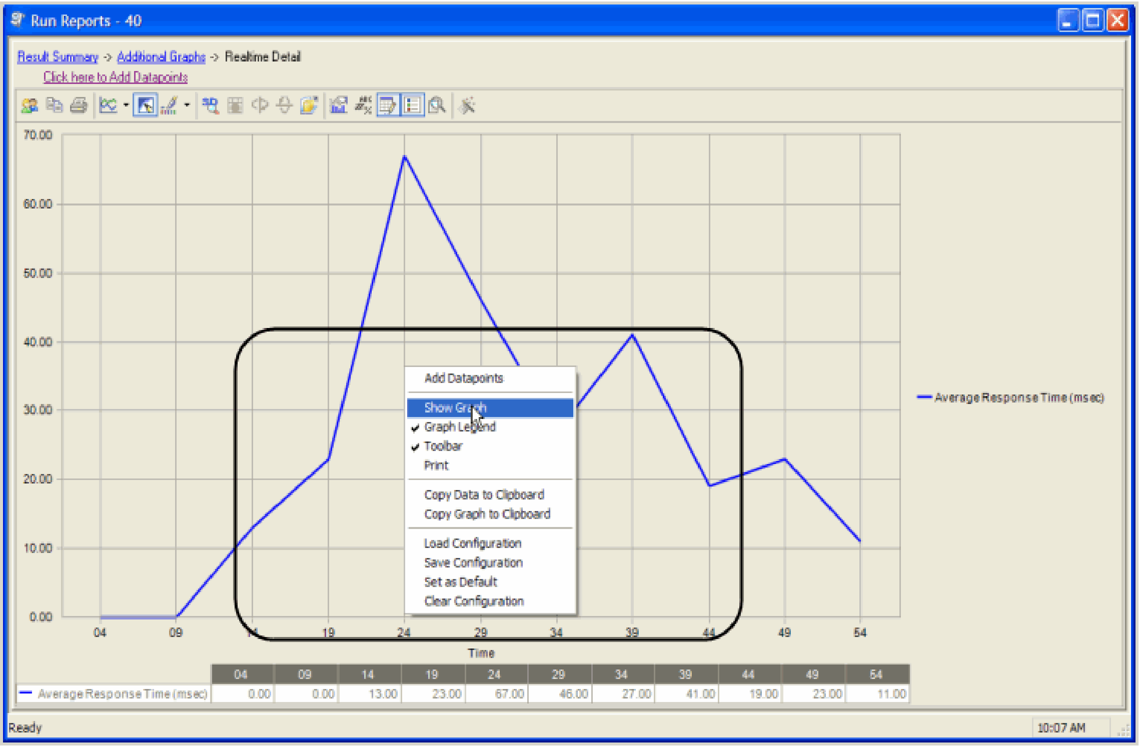
Saves a graph configuration

Sets the current configuration as default

Clears the current chart configuration

Show Data/Show Graph

Toggling to Graph displays the data in graph form.



Run Reports - 40

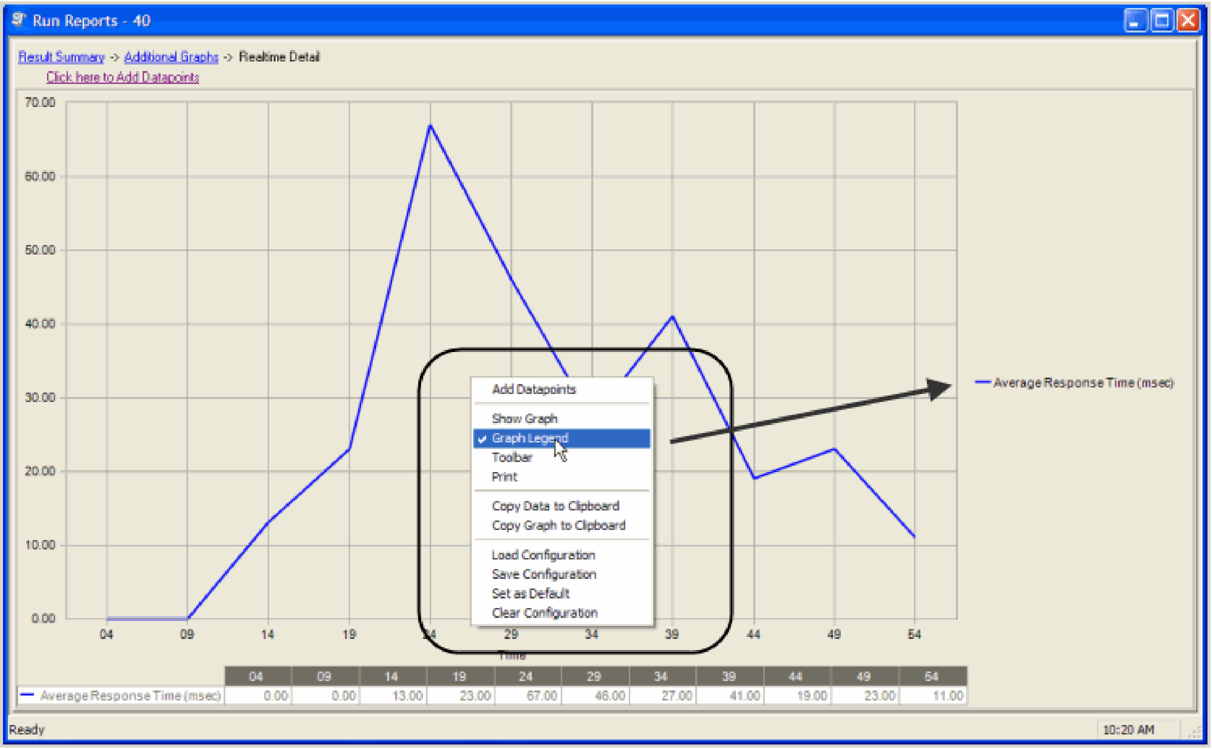
Result Summary -> Additional Graphs -> Realtime Detail

Click here to Add Datapoints

	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Graph Legend

Toggling to Graph Legend displays a legend on the side of the graph.



Run Reports - 40

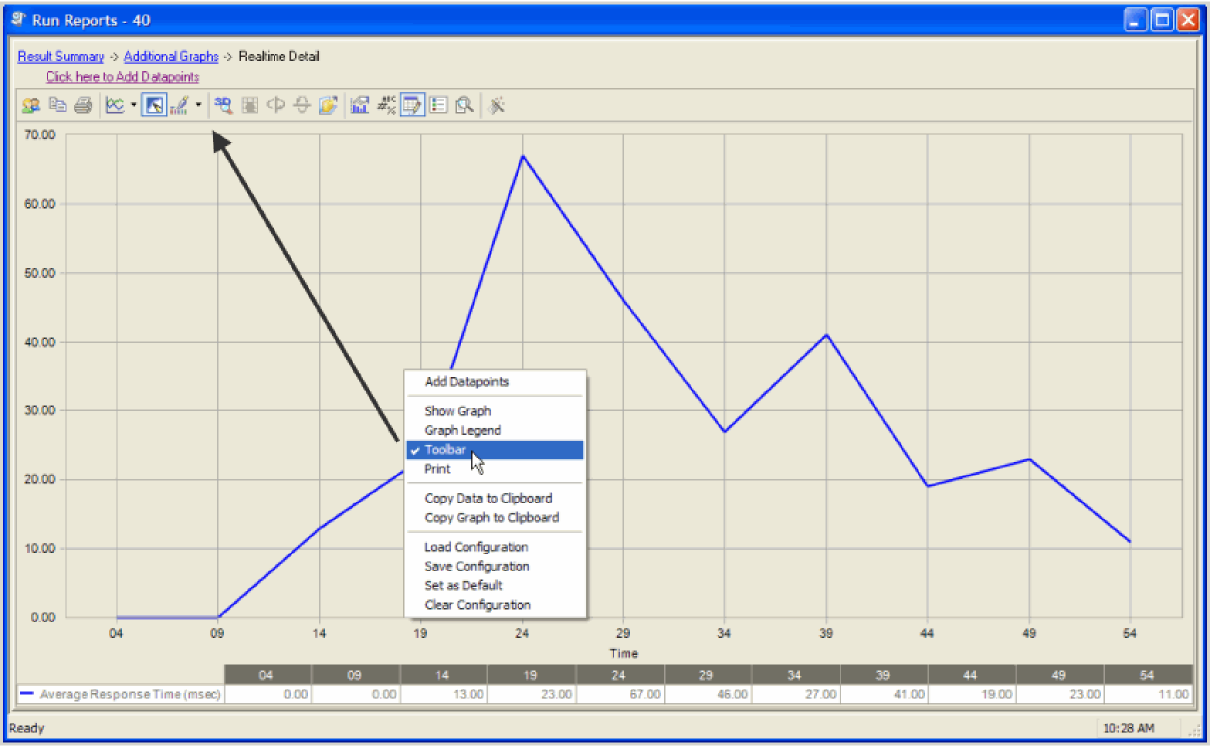
Result Summary -> Additional Graphs -> Realtime Detail

Click here to Add Datapoints

	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Toolbar

Toggling to Toolbar displays the graph toolbar.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Print

Choosing Print displays the Print Dialog.

Copy Data to Clipboard

Copies the data to the clipboard.

Copy Graph to Clipboard

Copies the graph to the clipboard.

Load Configuration

Benchmark factories graphs allows you to save graph configurations.

Save Configuration

Saves a graph configuration.

Set as Default

Sets a configured graph as default.

Clear Configuration

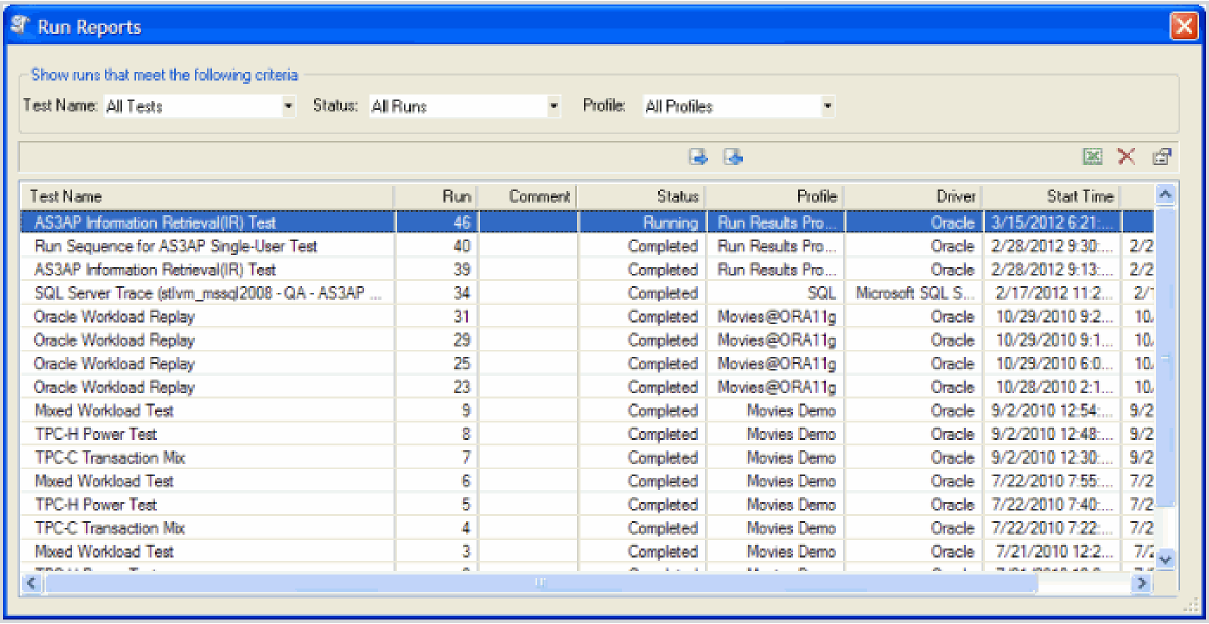
Clears a graph configuration.

Run Reports Viewer

Open the Run Reports viewer to view a list of all test result reports.

To open Run Reports

1. From the Benchmark Factory console click **Tools | Run Reports**. The Run Reports viewer opens.



Run Reports

Show runs that meet the following criteria

Test Name: All Tests

Status: All Runs

Profile: All Profiles

Test Name	Run	Comment	Status	Profile	Driver	Start Time	
AS3AP Information Retrieval(IR) Test	46		Running	Run Results Pro...	Oracle	3/15/2012 6:21:...	
Run Sequence for AS3AP Single-User Test	40		Completed	Run Results Pro...	Oracle	2/28/2012 9:30:...	2/2
AS3AP Information Retrieval(IR) Test	39		Completed	Run Results Pro...	Oracle	2/28/2012 9:13:...	2/2
SQL Server Trace (stlvm_mssql2008 - QA - AS3AP ...	34		Completed	SQL	Microsoft SQL S...	2/17/2012 11:2...	2/1
Oracle Workload Replay	31		Completed	Movies@ORA11g	Oracle	10/29/2010 9:2...	10.
Oracle Workload Replay	29		Completed	Movies@ORA11g	Oracle	10/29/2010 9:1...	10.
Oracle Workload Replay	25		Completed	Movies@ORA11g	Oracle	10/29/2010 6:0...	10.
Oracle Workload Replay	23		Completed	Movies@ORA11g	Oracle	10/28/2010 2:1...	10.
Mixed Workload Test	9		Completed	Movies Demo	Oracle	9/2/2010 12:54:...	9/2
TPC-H Power Test	8		Completed	Movies Demo	Oracle	9/2/2010 12:48:...	9/2
TPC-C Transaction Mix	7		Completed	Movies Demo	Oracle	9/2/2010 12:30:...	9/2
Mixed Workload Test	6		Completed	Movies Demo	Oracle	7/22/2010 7:55:...	7/2
TPC-H Power Test	5		Completed	Movies Demo	Oracle	7/22/2010 7:40:...	7/2
TPC-C Transaction Mix	4		Completed	Movies Demo	Oracle	7/22/2010 7:22:...	7/2
Mixed Workload Test	3		Completed	Movies Demo	Oracle	7/21/2010 12:2...	7/2

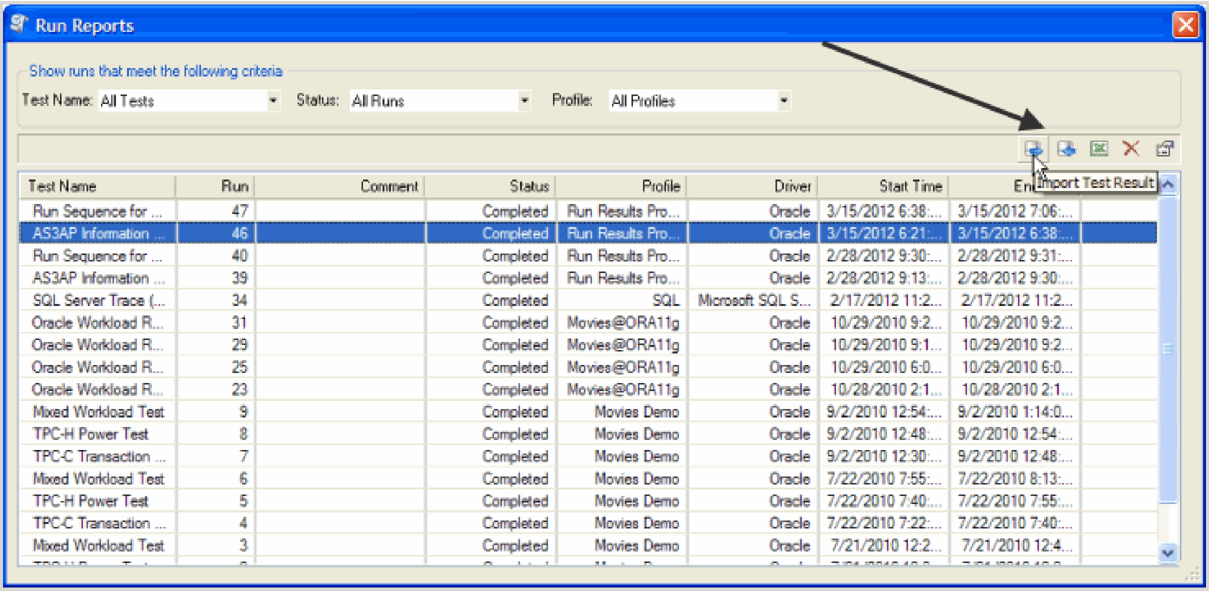
See Using Benchmark Factory Run Reports on page 180 for more information.

Importing and Exporting Benchmark Factory Test Results from Run Reports

From Run Reports, you can import or export run results.

To import or export testing results:

1. Open Run Reports.
2. Click on the required **Import** or **Export** button.



Show runs that meet the following criteria

Test Name: All Tests
Status: All Runs
Profile: All Profiles

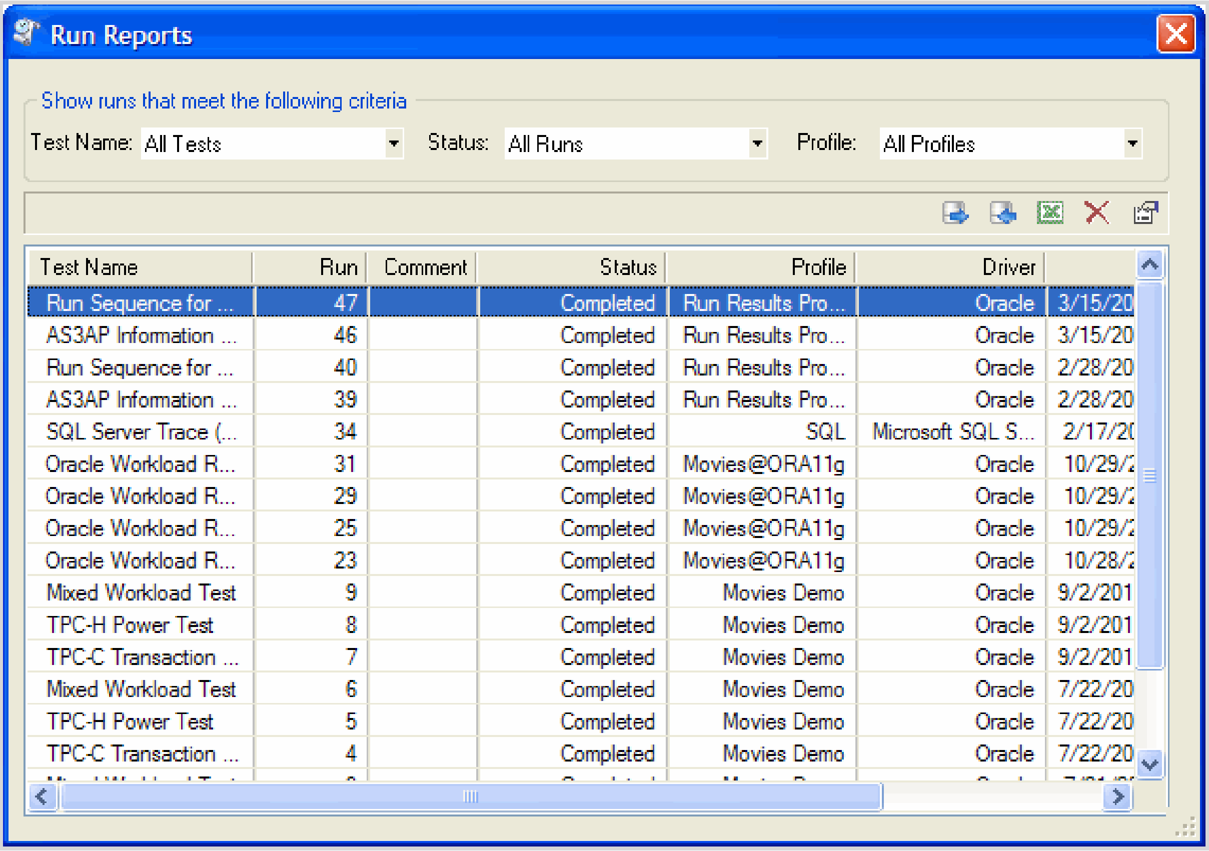
Test Name	Run	Comment	Status	Profile	Driver	Start Time	En
Run Sequence for ...	47		Completed	Run Results Pro...	Oracle	3/15/2012 6:38:...	3/15/2012 7:06:...
AS3AP Information ...	46		Completed	Run Results Pro...	Oracle	3/15/2012 6:21:...	3/15/2012 6:38:...
Run Sequence for ...	40		Completed	Run Results Pro...	Oracle	2/28/2012 9:30:...	2/28/2012 9:31:...
AS3AP Information ...	39		Completed	Run Results Pro...	Oracle	2/28/2012 9:13:...	2/28/2012 9:30:...
SQL Server Trace (...	34		Completed	SQL	Microsoft SQL S...	2/17/2012 11:2...	2/17/2012 11:2...
Oracle Workload R...	31		Completed	Movies@ORA11g	Oracle	10/29/2010 9:2...	10/29/2010 9:2...
Oracle Workload R...	29		Completed	Movies@ORA11g	Oracle	10/29/2010 9:1...	10/29/2010 9:2...
Oracle Workload R...	25		Completed	Movies@ORA11g	Oracle	10/29/2010 6:0...	10/29/2010 6:0...
Oracle Workload R...	23		Completed	Movies@ORA11g	Oracle	10/28/2010 2:1...	10/28/2010 2:1...
Mixed Workload Test	9		Completed	Movies Demo	Oracle	9/2/2010 12:54:...	9/2/2010 1:14:0...
TPC-H Power Test	8		Completed	Movies Demo	Oracle	9/2/2010 12:48:...	9/2/2010 12:54:...
TPC-C Transaction ...	7		Completed	Movies Demo	Oracle	9/2/2010 12:30:...	9/2/2010 12:48:...
Mixed Workload Test	6		Completed	Movies Demo	Oracle	7/22/2010 7:55:...	7/22/2010 8:13:...
TPC-H Power Test	5		Completed	Movies Demo	Oracle	7/22/2010 7:40:...	7/22/2010 7:55:...
TPC-C Transaction ...	4		Completed	Movies Demo	Oracle	7/22/2010 7:22:...	7/22/2010 7:40:...
Mixed Workload Test	3		Completed	Movies Demo	Oracle	7/21/2010 12:2...	7/21/2010 12:4...

Navigate to the required folder to import or export the test results.

3. Click **Open** or **Save**.

Using Benchmark Factory Run Reports

Benchmark Factory Run Reports provides a wide array of graphs, charts, and data that allow you to a analyze load testing scripts.



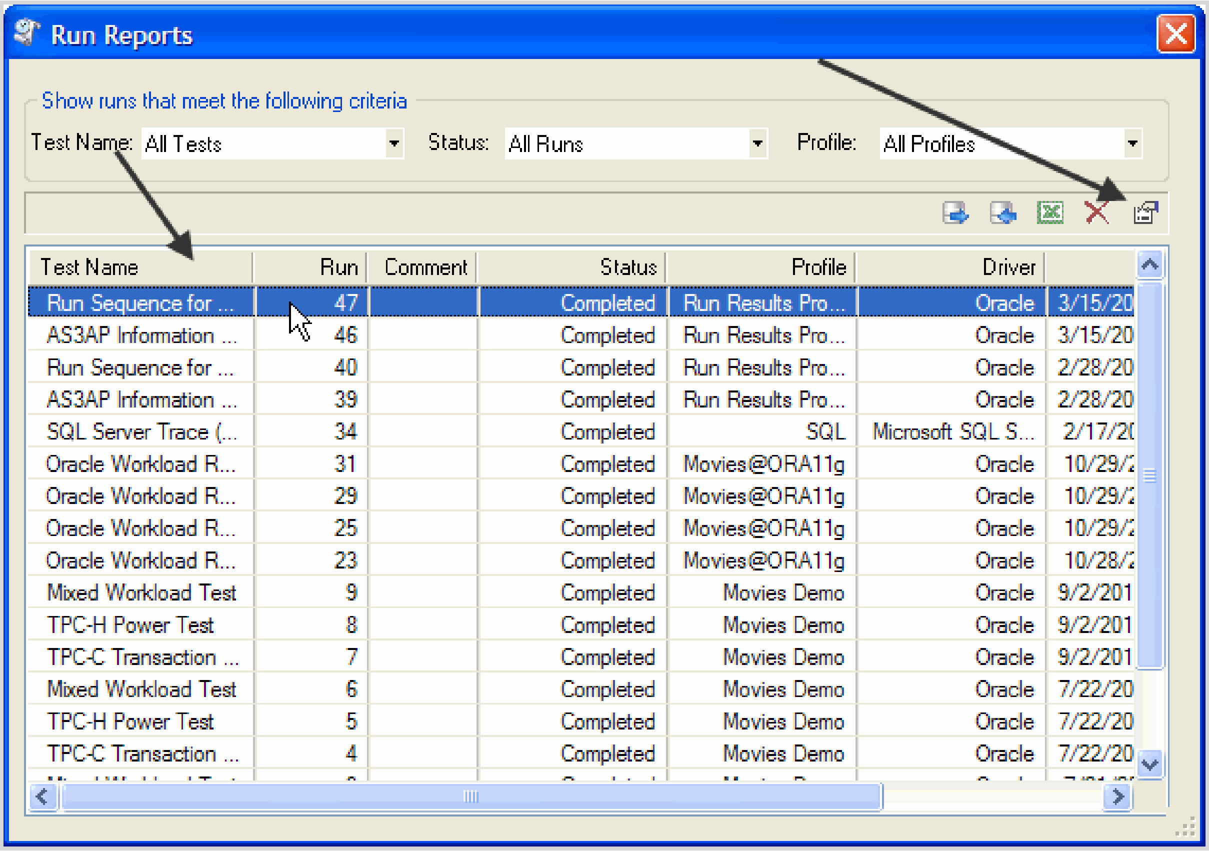
Show runs that meet the following criteria

Test Name: All Tests
Status: All Runs
Profile: All Profiles

Test Name	Run	Comment	Status	Profile	Driver	
Run Sequence for ...	47		Completed	Run Results Pro...	Oracle	3/15/20
AS3AP Information ...	46		Completed	Run Results Pro...	Oracle	3/15/20
Run Sequence for ...	40		Completed	Run Results Pro...	Oracle	2/28/20
AS3AP Information ...	39		Completed	Run Results Pro...	Oracle	2/28/20
SQL Server Trace (...	34		Completed	SQL	Microsoft SQL S...	2/17/20
Oracle Workload R...	31		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	29		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	25		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	23		Completed	Movies@ORA11g	Oracle	10/28/2
Mixed Workload Test	9		Completed	Movies Demo	Oracle	9/2/201
TPC-H Power Test	8		Completed	Movies Demo	Oracle	9/2/201
TPC-C Transaction ...	7		Completed	Movies Demo	Oracle	9/2/201
Mixed Workload Test	6		Completed	Movies Demo	Oracle	7/22/20
TPC-H Power Test	5		Completed	Movies Demo	Oracle	7/22/20
TPC-C Transaction ...	4		Completed	Movies Demo	Oracle	7/22/20

Navigating in Benchmark Factory Run Reports

To open a test, double-click on the test name, or select the test name and click the **Show Test Results** icon in the upper right corner of the dialog.

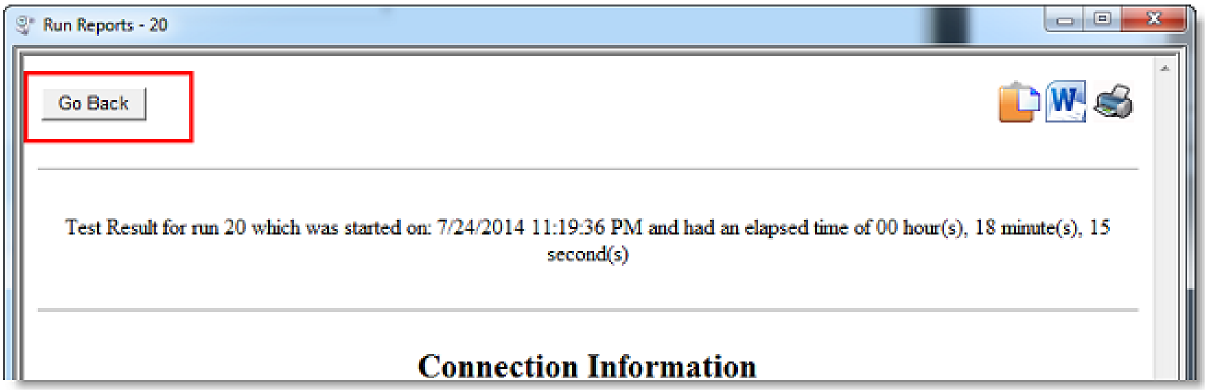


Show runs that meet the following criteria

Test Name: All Tests Status: All Runs Profile: All Profiles

Test Name	Run	Comment	Status	Profile	Driver	
Run Sequence for ...	47		Completed	Run Results Pro...	Oracle	3/15/20
AS3AP Information ...	46		Completed	Run Results Pro...	Oracle	3/15/20
Run Sequence for ...	40		Completed	Run Results Pro...	Oracle	2/28/20
AS3AP Information ...	39		Completed	Run Results Pro...	Oracle	2/28/20
SQL Server Trace (...	34		Completed	SQL	Microsoft SQL S...	2/17/20
Oracle Workload R...	31		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	29		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	25		Completed	Movies@ORA11g	Oracle	10/29/2
Oracle Workload R...	23		Completed	Movies@ORA11g	Oracle	10/28/2
Mixed Workload Test	9		Completed	Movies Demo	Oracle	9/2/201
TPC-H Power Test	8		Completed	Movies Demo	Oracle	9/2/201
TPC-C Transaction ...	7		Completed	Movies Demo	Oracle	9/2/201
Mixed Workload Test	6		Completed	Movies Demo	Oracle	7/22/20
TPC-H Power Test	5		Completed	Movies Demo	Oracle	7/22/20
TPC-C Transaction ...	4		Completed	Movies Demo	Oracle	7/22/20

When Benchmark Factory Run Reports opens, the Results Summary displays. Click **Go Back** to return to the previous screen.

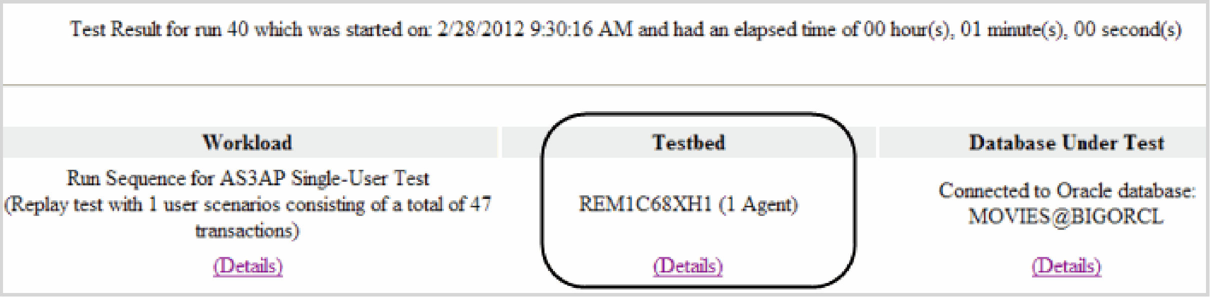


Test Result for run 20 which was started on: 7/24/2014 11:19:36 PM and had an elapsed time of 00 hour(s), 18 minute(s), 15 second(s)

Connection Information

To navigate Benchmark Factory Run Results

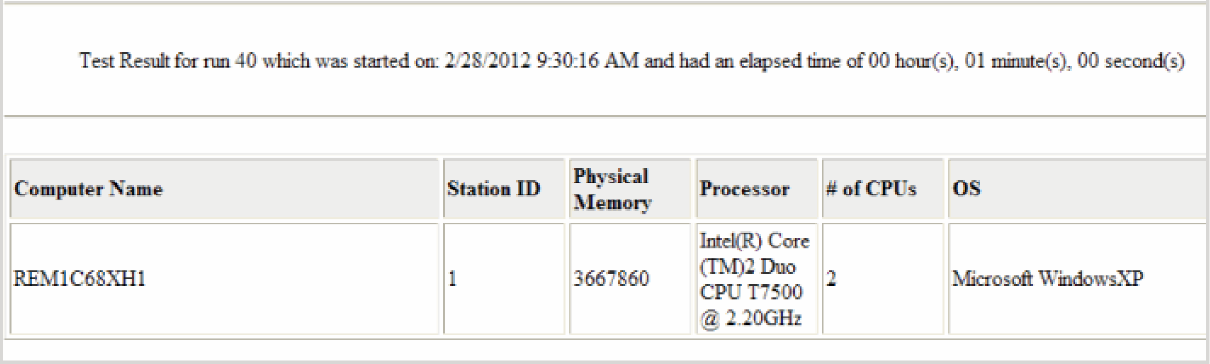
1. Select a section of the report to view, for example, **Testbed.**



Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Workload	Testbed	Database Under Test
Run Sequence for AS3AP Single-User Test (Replay test with 1 user scenarios consisting of a total of 47 transactions)	REM1C68XH1 (1 Agent)	Connected to Oracle database: MOVIES@BIGORCL
(Details)	(Details)	(Details)

2. Click **Details**. The Testbed information displays.



Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Computer Name	Station ID	Physical Memory	Processor	# of CPUs	OS
REM1C68XH1	1	3667860	Intel(R) Core (TM)2 Duo CPU T7500 @ 2.20GHz	2	Microsoft WindowsXP

3. To navigate back to Results Summary, click **Go Back** in the upper-left corner.

Run Reports provides the following views:

- Results Summary
- Workload
- Testbed
- Database Under Test

Results Summary

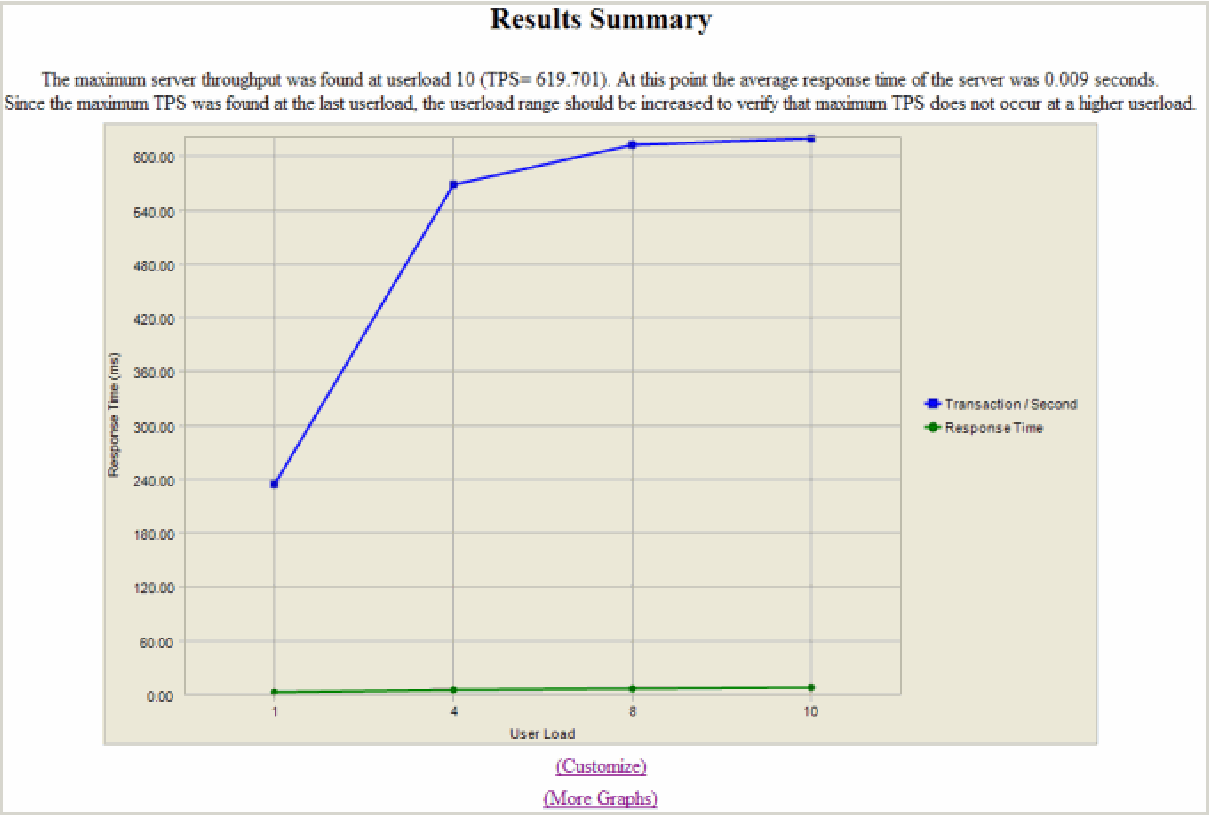
The Results Summary provides graphs and user load statistics for the selected test. The following graphs and tables are provided:

- Results Summary Graph
- Other Results Summary Graphs
- Using Benchmark Factory Run Reports

Results Summary Graph

Note: Not all tests provide the Results Summary graph.

This is a customizable graph that allows you to view selected data points.



Results Summary

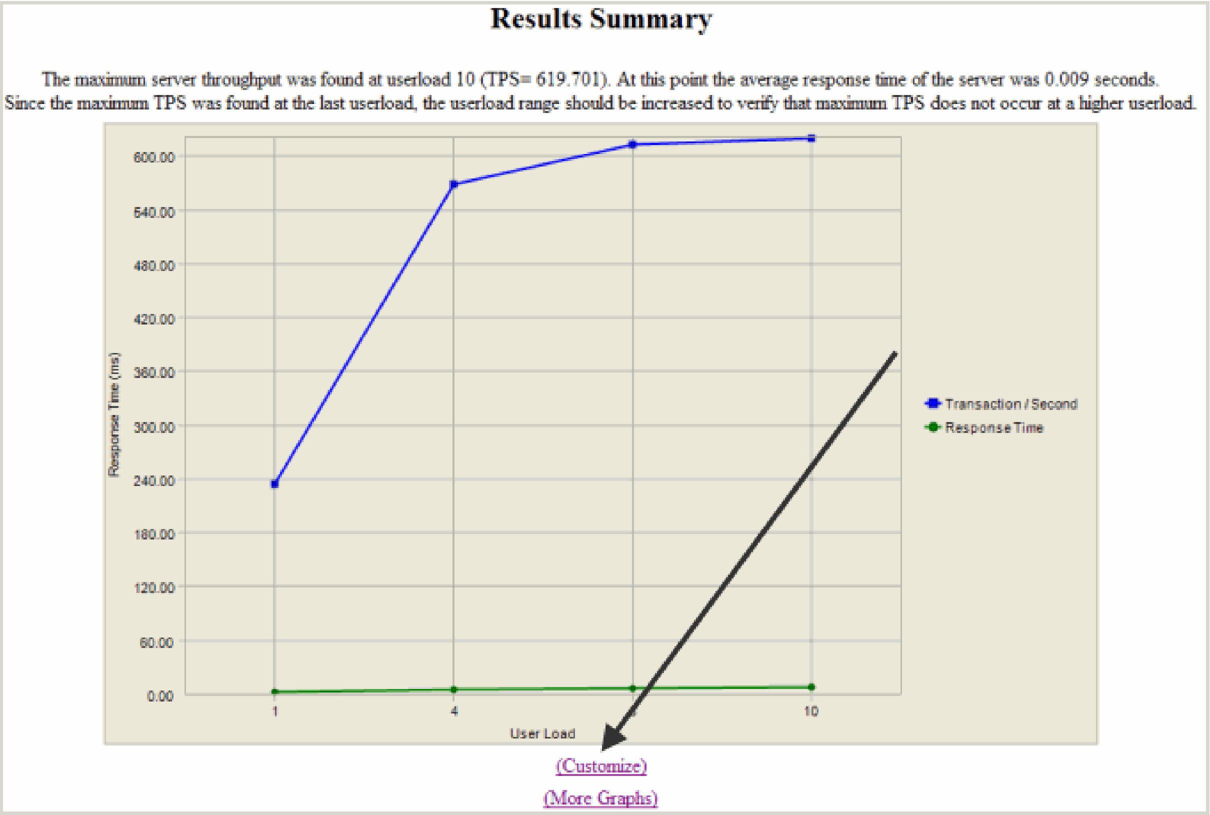
The maximum server throughput was found at userload 10 (TPS= 619.701). At this point the average response time of the server was 0.009 seconds. Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

(Customize)

(More Graphs)

To add or remove datapoints to the Results Summary graph

1. In the lower section of the graph, click **Customize**.



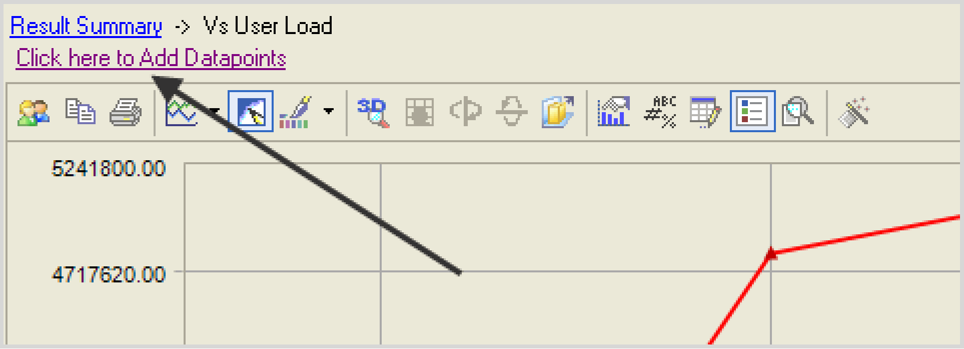
Results Summary

The maximum server throughput was found at userload 10 (TPS= 619.701). At this point the average response time of the server was 0.009 seconds. Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

(Customize)

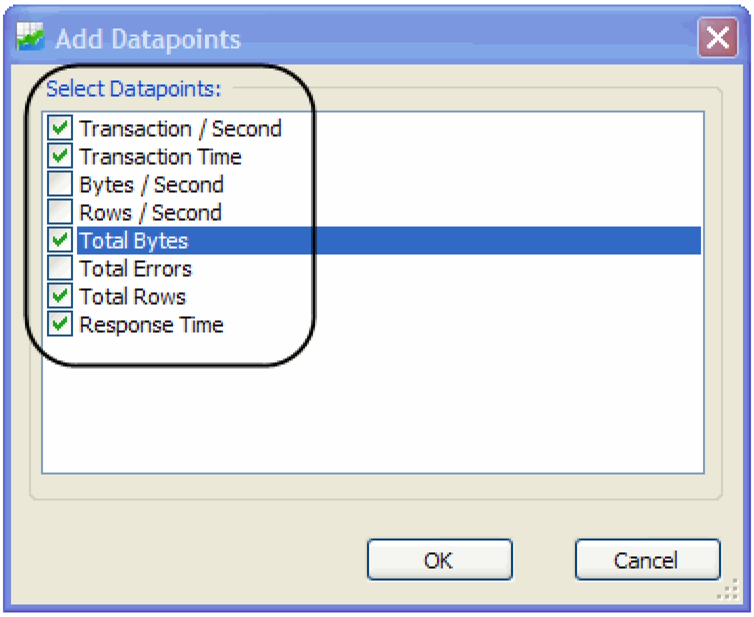
(More Graphs)

2. In the upper left corner, click **Click here to Add Datapoints**.



The Add Datapoints dialog displays.

2. Select the required datapoints.

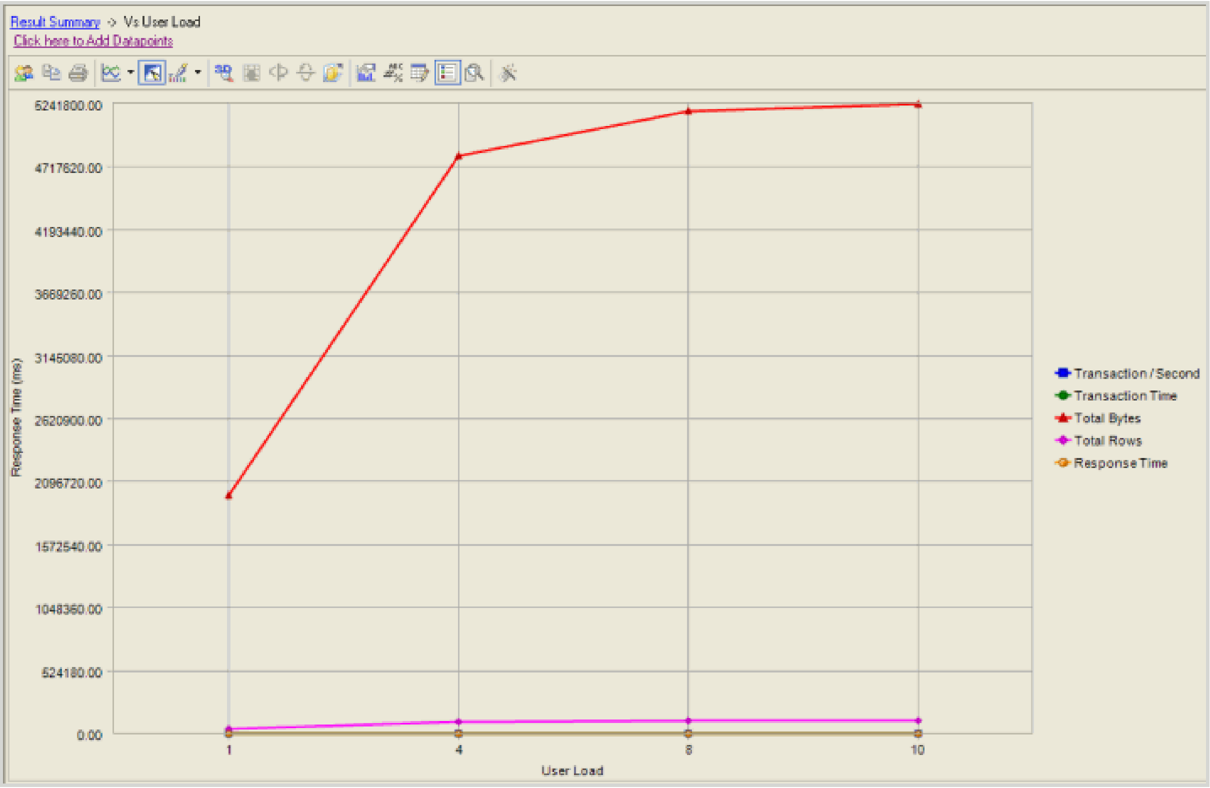


Select Datapoints:

☑ Transaction / Second
☑ Transaction Time
□ Bytes / Second
□ Rows / Second
☑ Total Bytes
□ Total Errors
☑ Total Rows
☑ Response Time

3. Click **Ok**.

   The selected datapoints display.



Result Summary -> Vs User Load

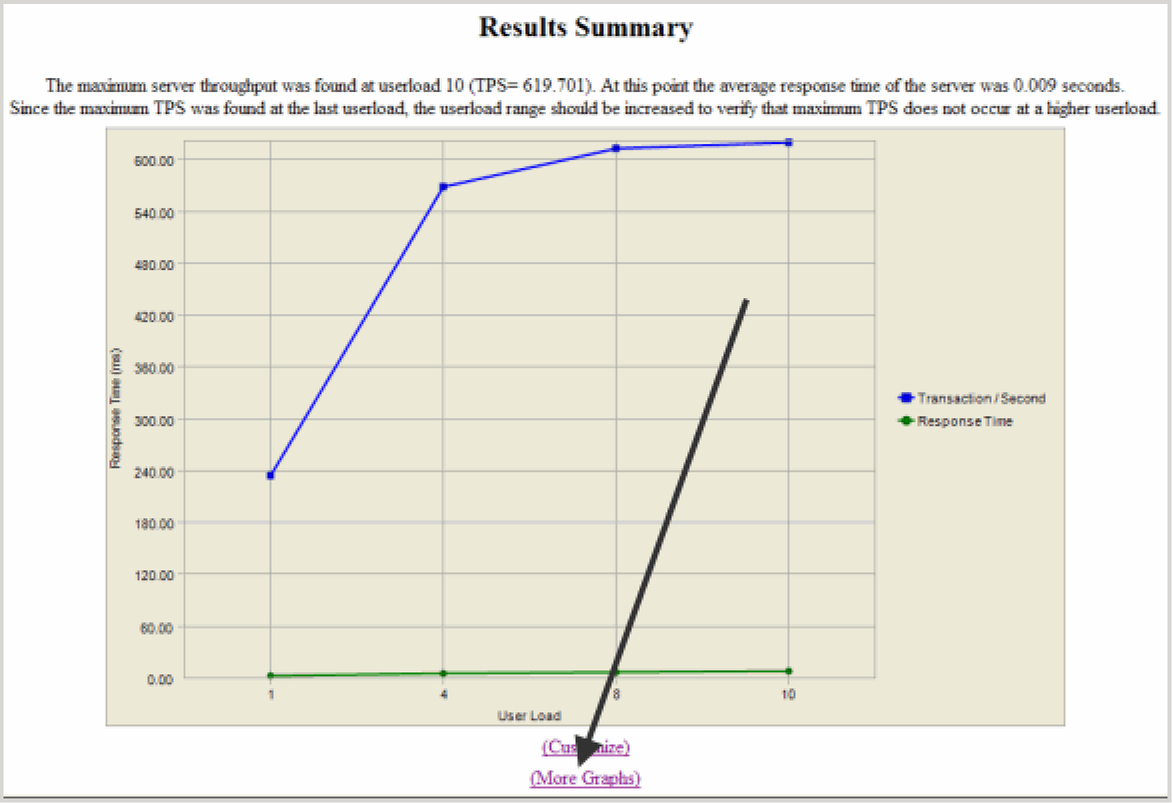
Click here to Add Datapoints

Other Results Summary Graphs

The Results Summary view allows you to view other graphs that provide test details.

To view other Results Summary Graphs

1. In the Results Summary view, click **More Graphs**.



Results Summary

The maximum server throughput was found at userload 10 (TPS= 619.701). At this point the average response time of the server was 0.009 seconds. Since the maximum TPS was found at the last userload, the userload range should be increased to verify that maximum TPS does not occur at a higher userload.

The following graphs display:

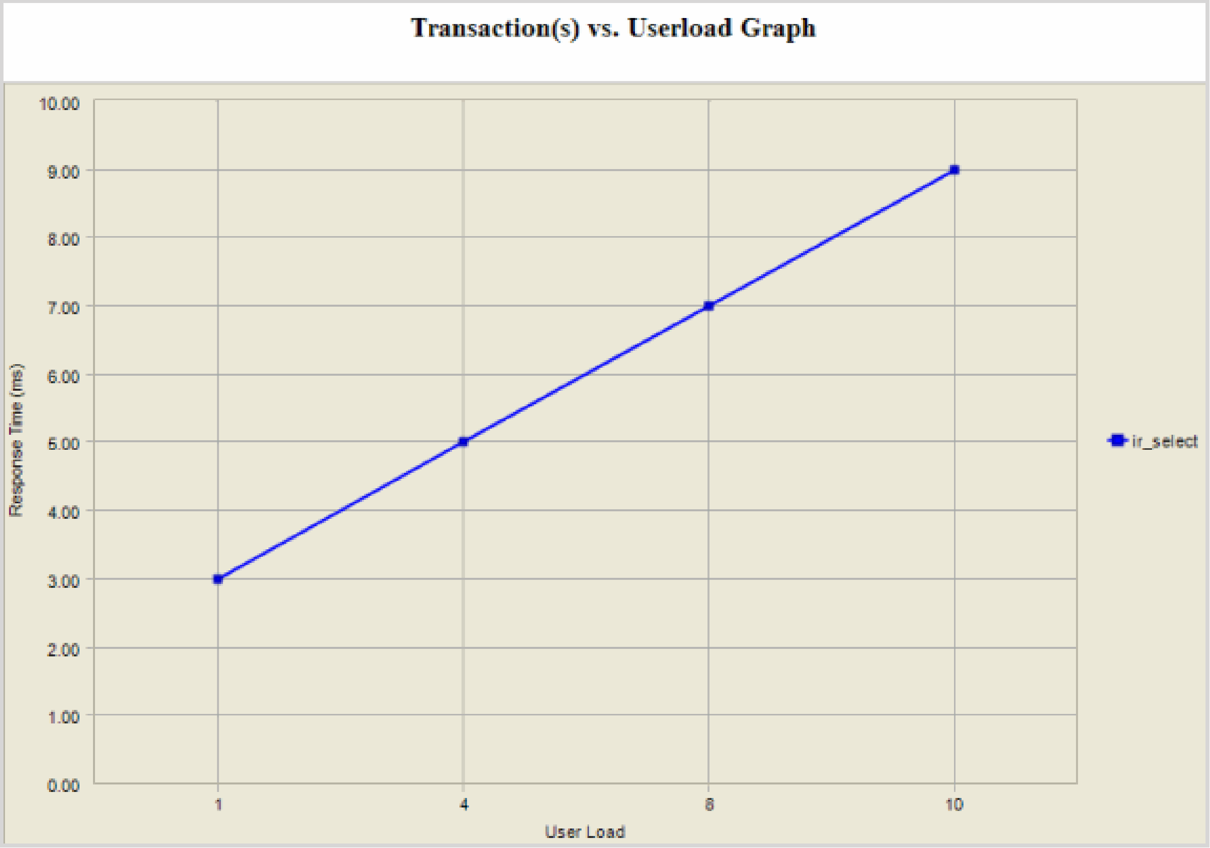
- Userload
- Realtime Summary
- Realtime Detail

Userload

Understanding how user loads affect the performance of a database is essential to end user satisfaction. The Userload graph plots user load against:

- Response Time
- Total Bytes
- Total Errors
- Total Rows
- Response Time

Reviewing these datapoints allows you to fully understand how "real-world" userloads affect database performance.

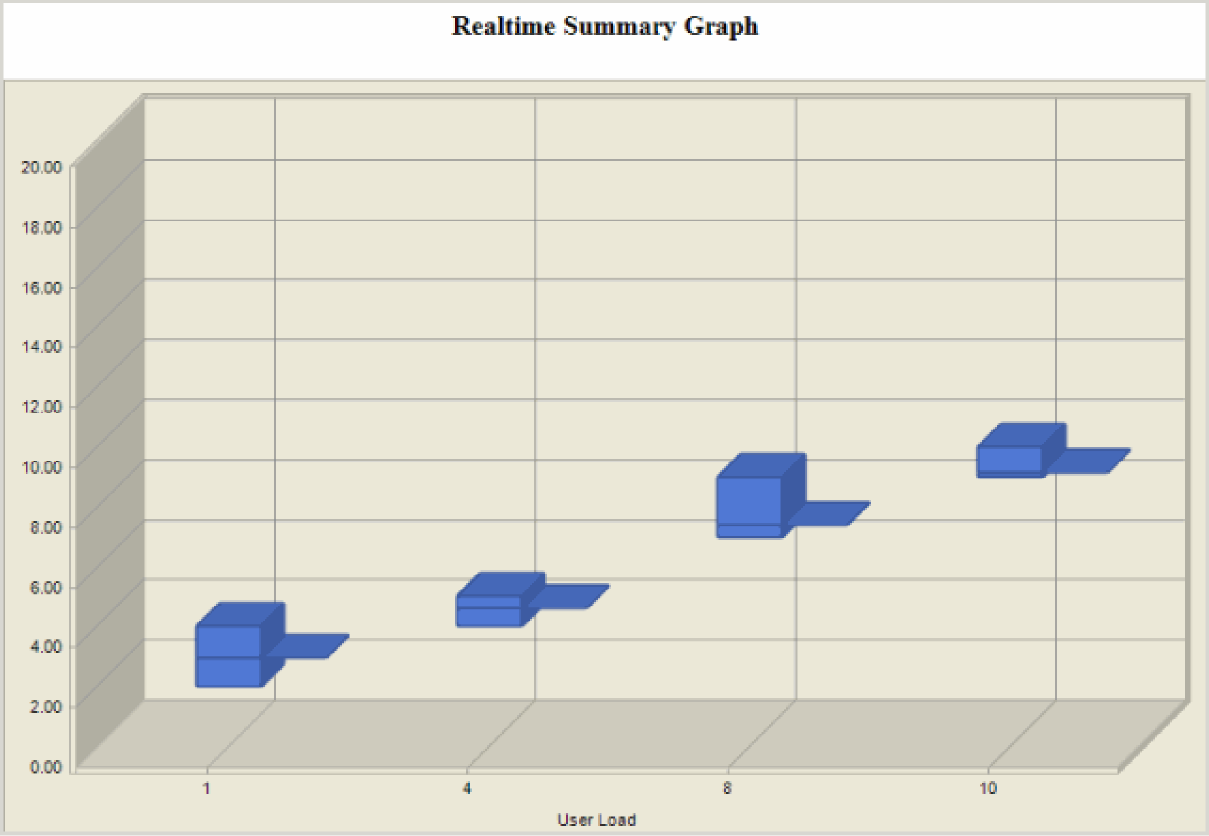


Realtime Summary

The Realtime Statistic graph allow you to view what "actually" happened during a load test. You can plot userload against:

- Transaction/Second
- Total Rows
- Errors

- Total Bytes
- Rows/Second
- Minimum Time
- Maximum Response Time
- Deadlocks
- Bytes/Second
- Average Time
- Bytes/Second

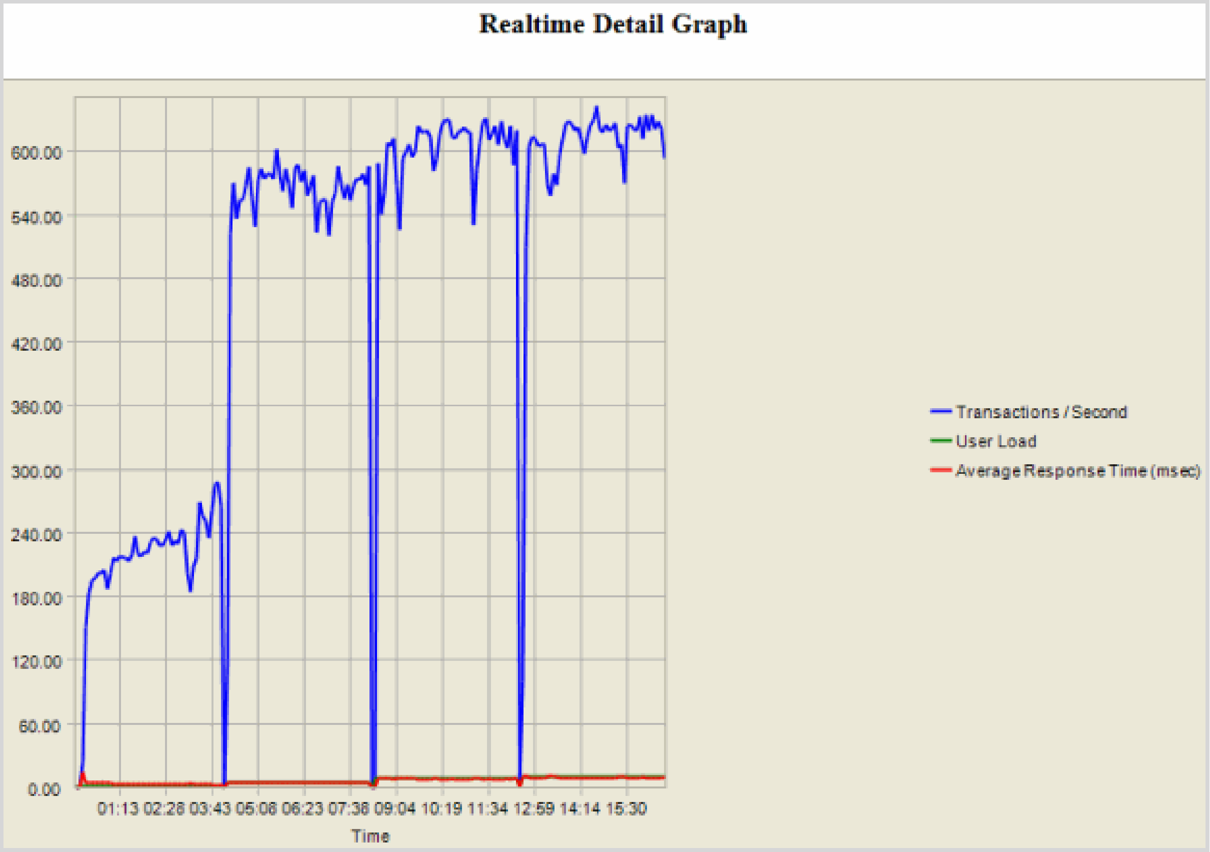


Realtime Detail

The Realtime Detail graph allows you to view what actually happened during the running of a load test. This allows you to view the actual timing events. From the Realtime Detail graph you can view:

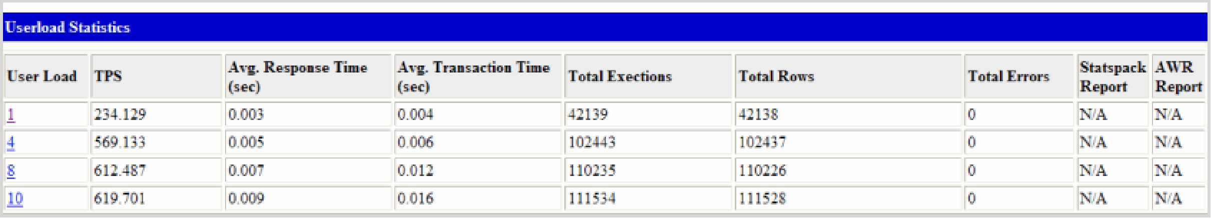
- Average Response Time
- Average Time
- Bytes/ Second
- Deadlocks
- Maximum Response Time
- Maximum Time
- Minimum Response Time
- Minimum Time
- Rows/Second

- Total Bytes
- Errors
- Total Rows
- Transactions/Second
- User Load



Userload Statistics Table

This table is a snapshot of the workload test. You to quickly review a complete overview of database that includes, userload, transactions per second (TPS), average response time (sec).



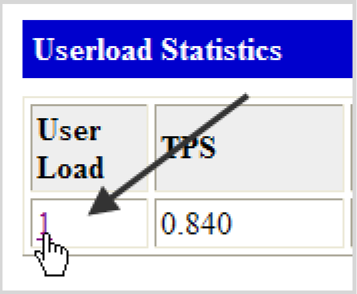
Userload Statistics

User Load	TPS	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Total Exections	Total Rows	Total Errors	Statspack Report	AWR Report
1	234.129	0.003	0.004	42139	42138	0	N/A	N/A
4	569.133	0.005	0.006	102443	102437	0	N/A	N/A
8	612.487	0.007	0.012	110235	110226	0	N/A	N/A
10	619.701	0.009	0.016	111534	111528	0	N/A	N/A

From this table you can drill down to view userload stastics.

To drill down from the Userload Statistics table

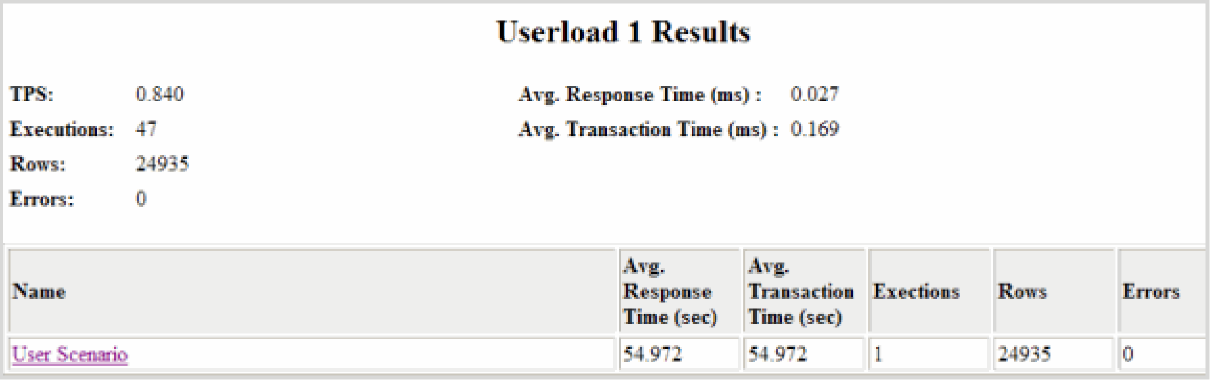
1. Click on the required Userload.



Userload Statistics

User Load	TPS
1	0.840

The Userload Results view displays.

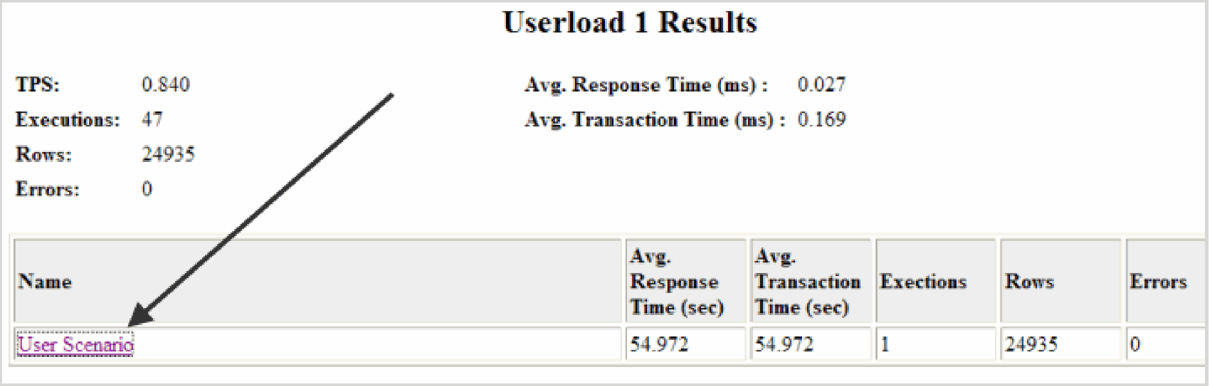


Userload 1 Results

TPS:	0.840	**Avg. Response Time (ms) :**	0.027
Executions:	47	**Avg. Transaction Time (ms) :**	0.169
Rows:	24935		
Errors:	0		

Name	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Exections	Rows	Errors
User Scenario	54.972	54.972	1	24935	0

2. From this view, you drill down further by clicking on required User Scenario name.

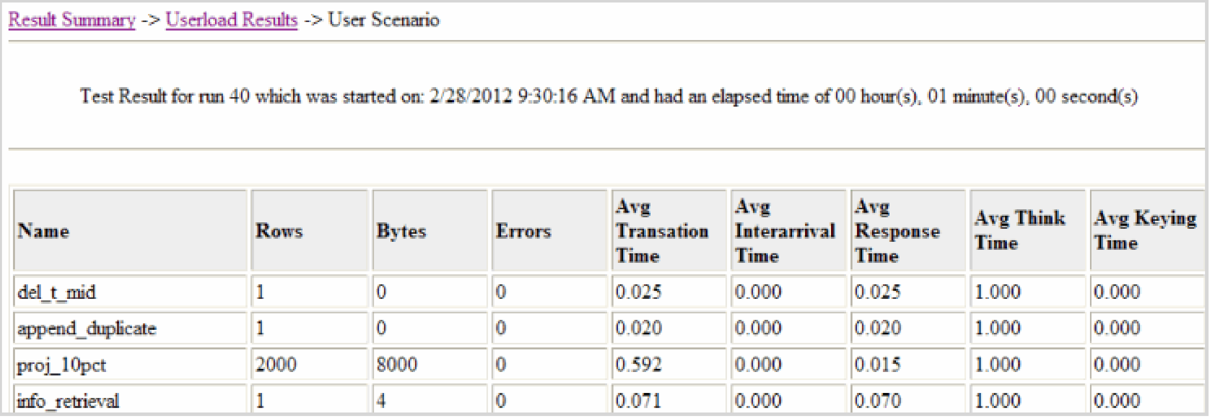


Userload 1 Results

TPS:	0.840	**Avg. Response Time (ms) :**	0.027
Executions:	47	**Avg. Transaction Time (ms) :**	0.169
Rows:	24935		
Errors:	0		

Name	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Exections	Rows	Errors
User Scenario	54.972	54.972	1	24935	0

This displays detailed test results broken down by row name.



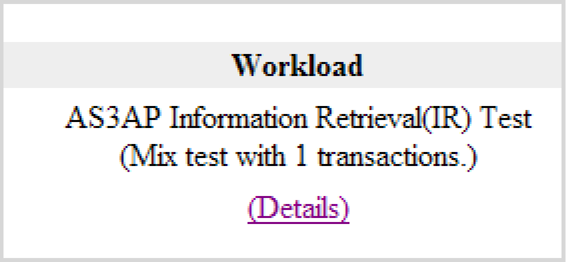
Result Summary -> Userload Results -> User Scenario

Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Name	Rows	Bytes	Errors	Avg Transation Time	Avg Interarrival Time	Avg Response Time	Avg Think Time	Avg Keying Time
del_t_mid	1	0	0	0.025	0.000	0.025	1.000	0.000
append_duplicate	1	0	0	0.020	0.000	0.020	1.000	0.000
proj_10pct	2000	8000	0	0.592	0.000	0.015	1.000	0.000
info_retrieval	1	4	0	0.071	0.000	0.070	1.000	0.000

Workload

The Workload view provides testing details.

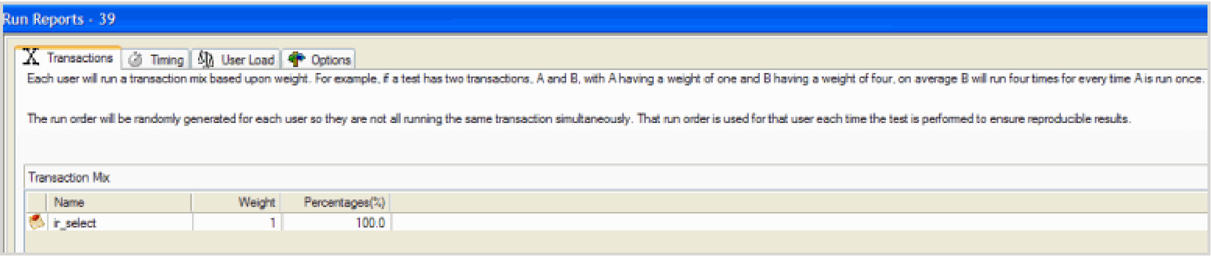


Workload

AS3AP Information Retrieval(IR) Test (Mix test with 1 transactions.)

(Details)

Clicking **Details** displays the transactions mix, timing, userload, and testing options used for the test.



Run Reports - 39

Each user will run a transaction mix based upon weight. For example, if a test has two transactions, A and B, with A having a weight of one and B having a weight of four, on average B will run four times for every time A is run once.

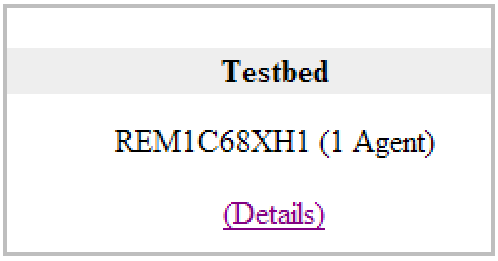
The run order will be randomly generated for each user so they are not all running the same transaction simultaneously. That run order is used for that user each time the test is performed to ensure reproducible results.

Transaction Mix

Name	Weight	Percentages(%)
ir_select	1	100.0

Testbed

The Testbed view shows data on agent configuration and processes during the running of a job.

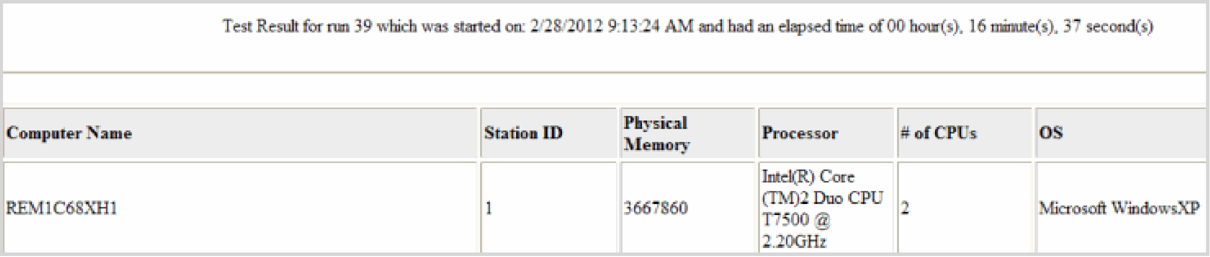


Testbed

REM1C68XH1 (1 Agent)

(Details)

Clicking **Details** displays a table with machine and operating system details.

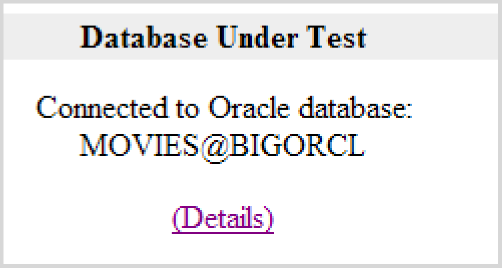


Test Result for run 39 which was started on: 2/28/2012 9:13:24 AM and had an elapsed time of 00 hour(s), 16 minute(s), 37 second(s)

Computer Name	Station ID	Physical Memory	Processor	# of CPUs	OS
REM1C68XH1	1	3667860	Intel(R) Core (TM)2 Duo CPU T7500 @ 2.20GHz	2	Microsoft WindowsXP

Database Under Test

This view displays connection and database information.

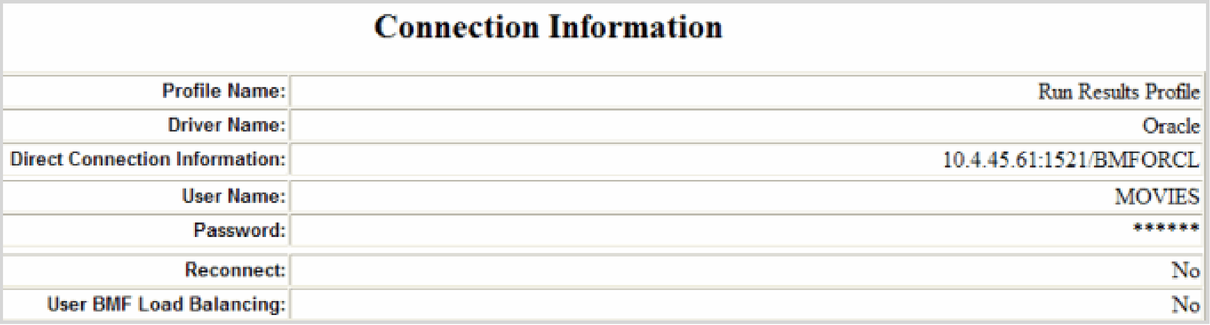


Database Under Test

Connected to Oracle database: MOVIES@BIGORCL

(Details)

Click **Details** to display connection and database information details.



Connection Information

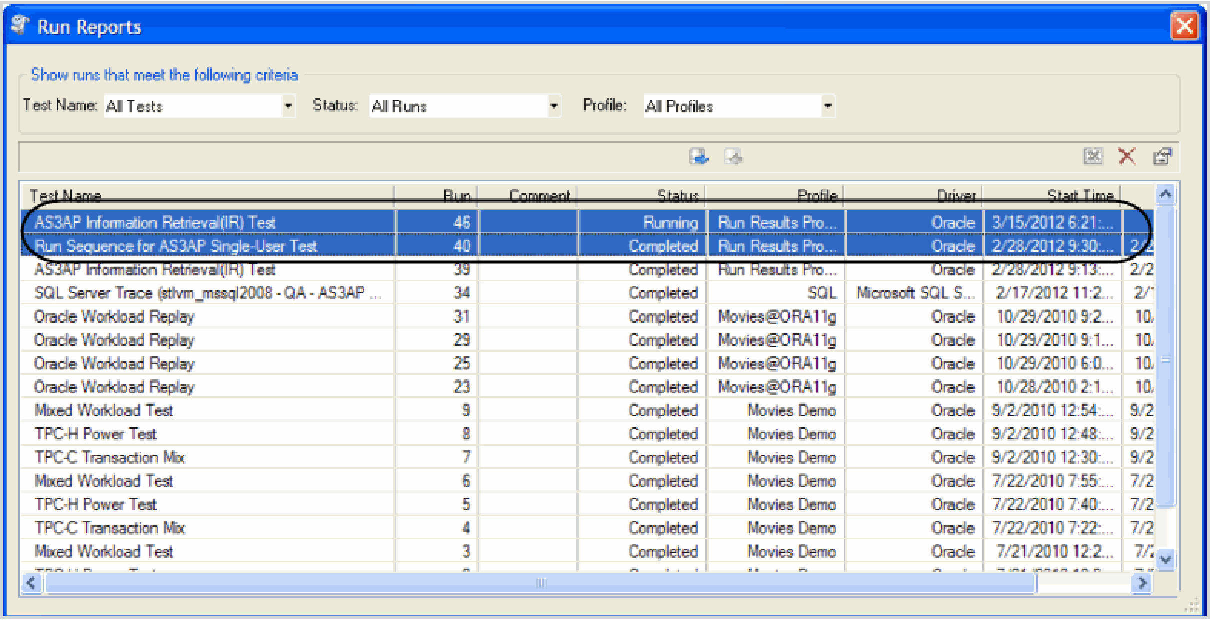
Profile Name:	Run Results Profile
Driver Name:	Oracle
Direct Connection Information:	10.4.45.61:1521/BMFORCL
User Name:	MOVIES
Password:	******
Reconnect:	No
User BMF Load Balancing:	No

View Multiple Run Results

Benchmark Factory allows you to view and compare multiple results using Run Reports.

To view multiple Run Reports results

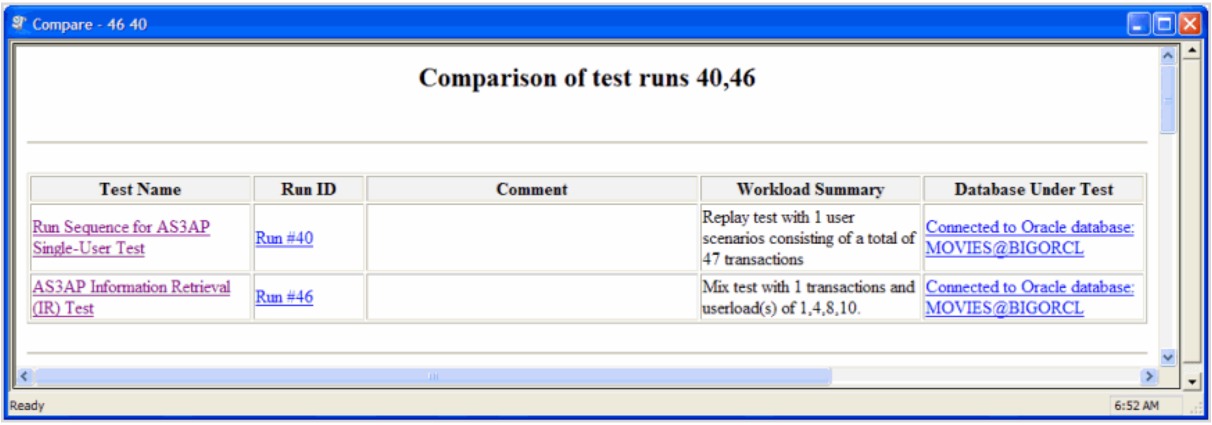
1. Select the desired runs.



Test Name: All Tests
Status: All Runs
Profile: All Profiles

Test Name	Run	Comment	Status	Profile	Driver	Start Time	
AS3AP Information Retrieval(IR) Test	46		Running	Run Results Pro...	Oracle	3/15/2012 6:21:...	
Run Sequence for AS3AP Single-User Test	40		Completed	Run Results Pro...	Oracle	2/28/2012 9:30:...	2/2
AS3AP Information Retrieval(IR) Test	39		Completed	Run Results Pro...	Oracle	2/28/2012 9:13:...	2/2
SQL Server Trace (stlvm_mssql2008 - QA - AS3AP ...	34		Completed	SQL	Microsoft SQL S...	2/17/2012 11:2...	2/1
Oracle Workload Replay	31		Completed	Movies@ORA11g	Oracle	10/29/2010 9:2...	10/
Oracle Workload Replay	29		Completed	Movies@ORA11g	Oracle	10/29/2010 9:1...	10/
Oracle Workload Replay	25		Completed	Movies@ORA11g	Oracle	10/29/2010 6:0...	10/
Oracle Workload Replay	23		Completed	Movies@ORA11g	Oracle	10/28/2010 2:1...	10/
Mixed Workload Test	9		Completed	Movies Demo	Oracle	9/2/2010 12:54:...	9/2
TPC-H Power Test	8		Completed	Movies Demo	Oracle	9/2/2010 12:48:...	9/2
TPC-C Transaction Mix	7		Completed	Movies Demo	Oracle	9/2/2010 12:30:...	9/2
Mixed Workload Test	6		Completed	Movies Demo	Oracle	7/22/2010 7:55:...	7/2
TPC-H Power Test	5		Completed	Movies Demo	Oracle	7/22/2010 7:40:...	7/2
TPC-C Transaction Mix	4		Completed	Movies Demo	Oracle	7/22/2010 7:22:...	7/2
Mixed Workload Test	3		Completed	Movies Demo	Oracle	7/21/2010 12:2...	7/2

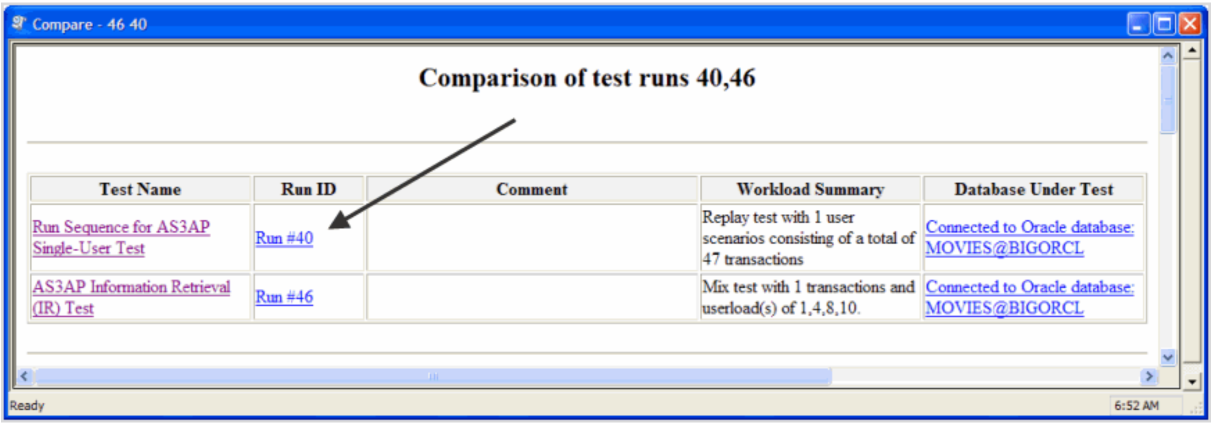
2. Click the **Show Test Results** icon in the upper right corner of the dialog. Run Reports displays



Comparison of test runs 40,46

Test Name	Run ID	Comment	Workload Summary	Database Under Test
Run Sequence for AS3AP Single-User Test	Run #40		Replay test with 1 user scenarios consisting of a total of 47 transactions	Connected to Oracle database: MOVIES@BIGORCL
AS3AP Information Retrieval (IR) Test	Run #46		Mix test with 1 transactions and userload(s) of 1,4,8,10.	Connected to Oracle database: MOVIES@BIGORCL

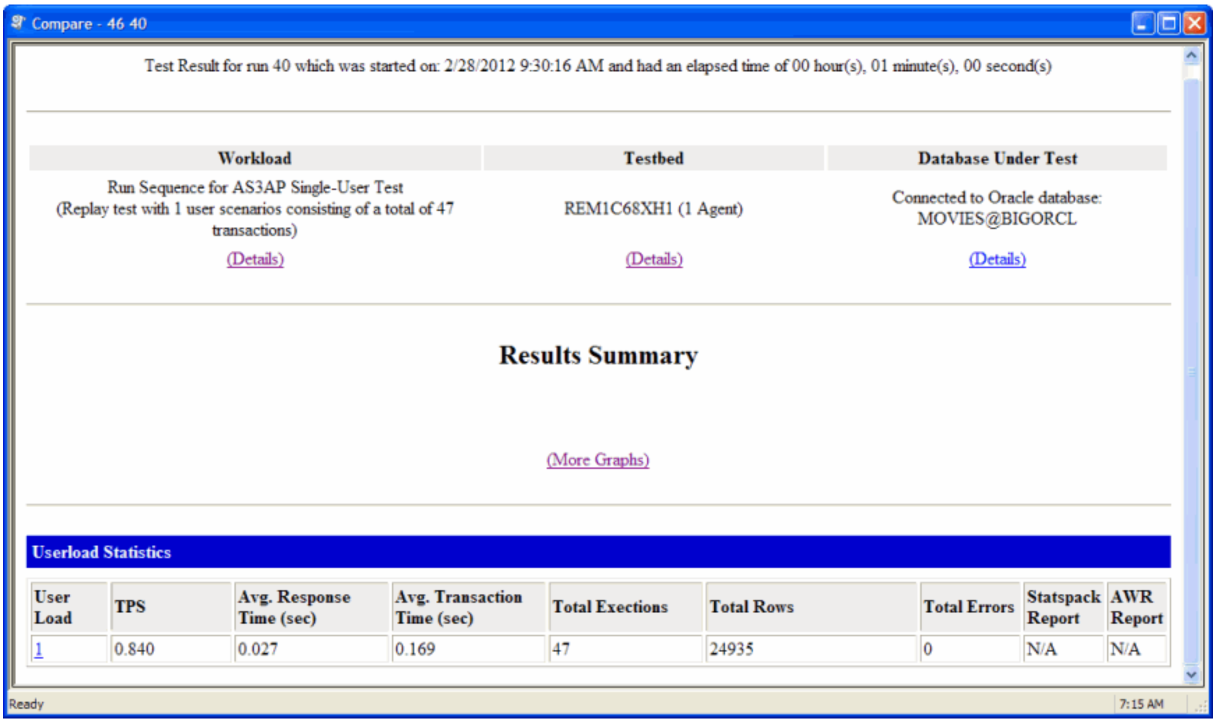
3. Click on an individual test run to view those results.



Comparison of test runs 40,46

Test Name	Run ID	Comment	Workload Summary	Database Under Test
Run Sequence for AS3AP Single-User Test	Run #40		Replay test with 1 user scenarios consisting of a total of 47 transactions	Connected to Oracle database: MOVIES@BIGORCL
AS3AP Information Retrieval (IR) Test	Run #46		Mix test with 1 transactions and userload(s) of 1,4,8,10.	Connected to Oracle database: MOVIES@BIGORCL

The results for that test display.



Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Workload	Testbed	Database Under Test
Run Sequence for AS3AP Single-User Test (Replay test with 1 user scenarios consisting of a total of 47 transactions) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

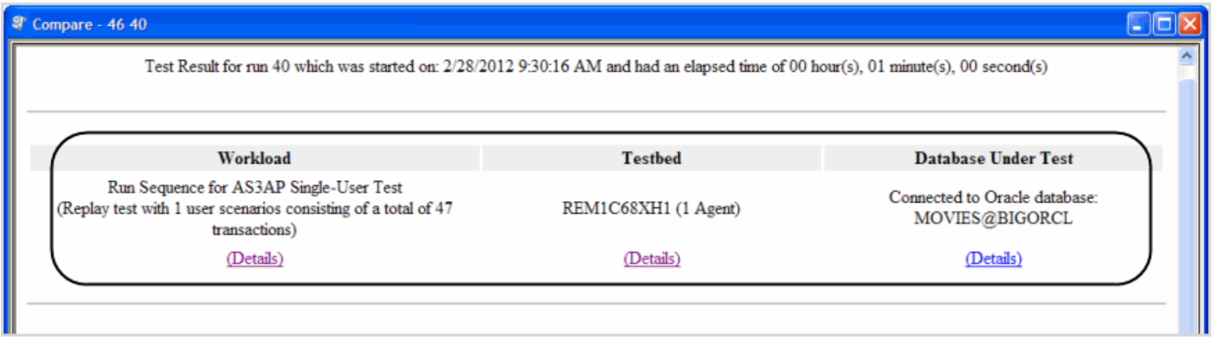
Results Summary

(More Graphs)

Userload Statistics

User Load	TPS	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Total Exections	Total Rows	Total Errors	Statspack Report	AWR Report
1	0.840	0.027	0.169	47	24935	0	N/A	N/A

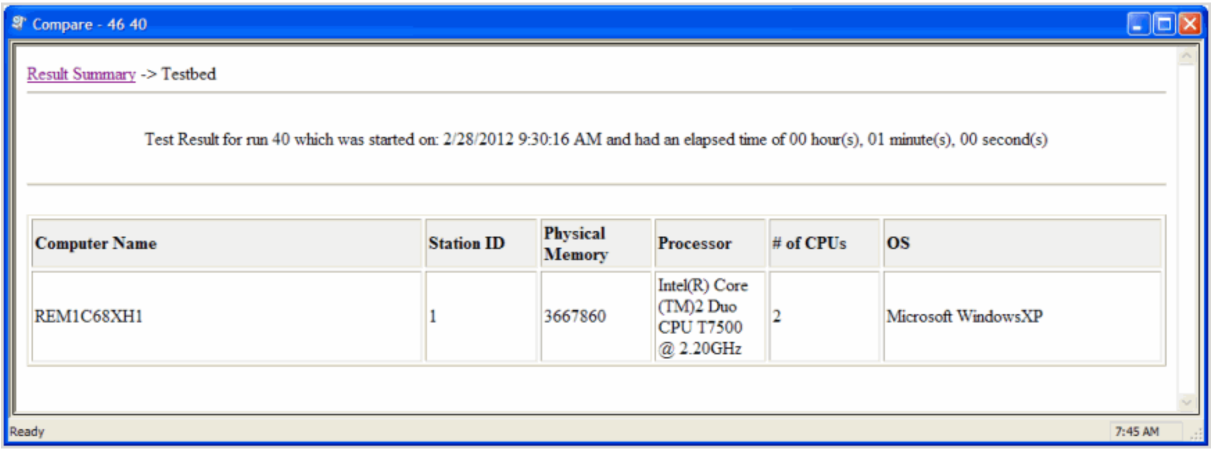
4. Click on **Details** for **Workload**, **Testbed**, or **Database Under Test** to drill down on testing results.



Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Workload	Testbed	Database Under Test
Run Sequence for AS3AP Single-User Test (Replay test with 1 user scenarios consisting of a total of 47 transactions) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

The following graphic shows Testbed drill down results.



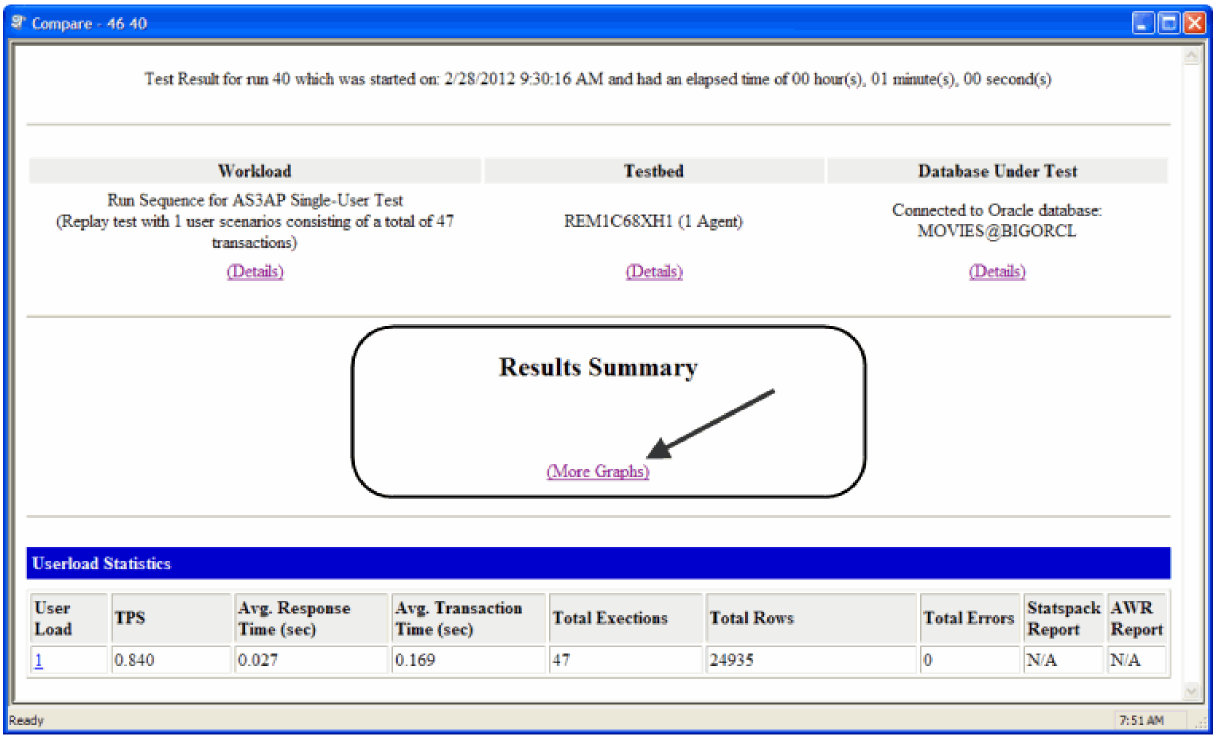
Compare - 46 40

Result Summary -> Testbed

Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Computer Name	Station ID	Physical Memory	Processor	# of CPUs	OS
REM1C68XH1	1	3667860	Intel(R) Core (TM)2 Duo CPU T7500 @ 2.20GHz	2	Microsoft WindowsXP

5. Click **More Graphs** in Results Summary.



Compare - 46 40

Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Workload	Testbed	Database Under Test
Run Sequence for AS3AP Single-User Test (Replay test with 1 user scenarios consisting of a total of 47 transactions) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

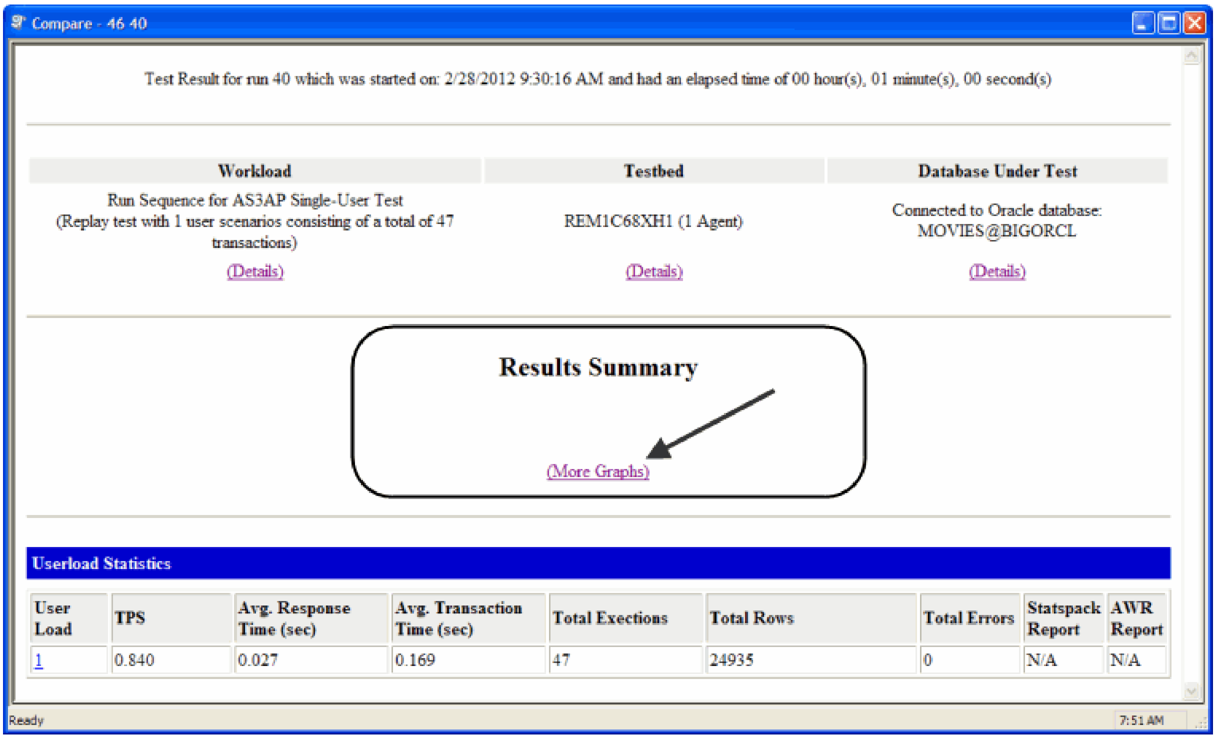
Results Summary

(More Graphs)

Userload Statistics

User Load	TPS	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Total Exections	Total Rows	Total Errors	Statspack Report	AWR Report
1	0.840	0.027	0.169	47	24935	0	N/A	N/A

The Results Summary graphs display.



Compare - 46 40

Test Result for run 40 which was started on: 2/28/2012 9:30:16 AM and had an elapsed time of 00 hour(s), 01 minute(s), 00 second(s)

Workload	Testbed	Database Under Test
Run Sequence for AS3AP Single-User Test (Replay test with 1 user scenarios consisting of a total of 47 transactions) (Details)	REM1C68XH1 (1 Agent) (Details)	Connected to Oracle database: MOVIES@BIGORCL (Details)

Results Summary

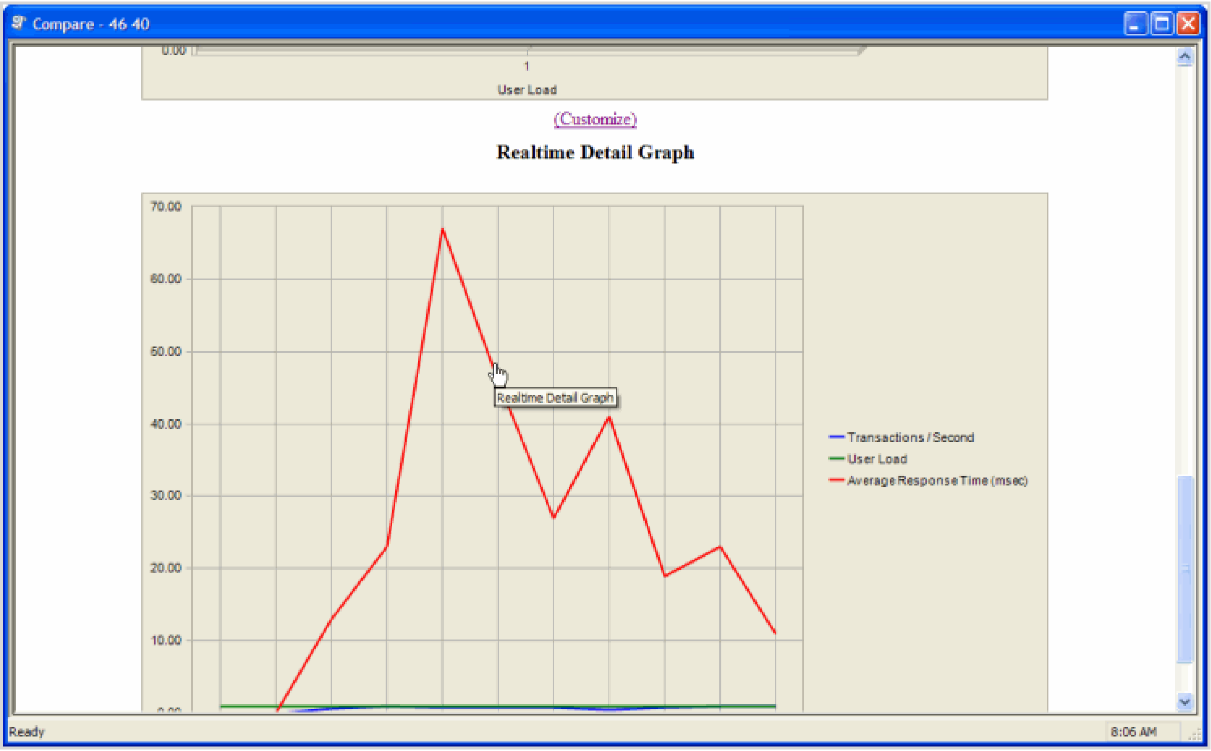
(More Graphs)

Userload Statistics

User Load	TPS	Avg. Response Time (sec)	Avg. Transaction Time (sec)	Total Exections	Total Rows	Total Errors	Statspack Report	AWR Report
1	0.840	0.027	0.169	47	24935	0	N/A	N/A

Ready 7:51 AM

View other graphs as required.



Compare - 46 40

(Customize)

Realtime Detail Graph

6. Close Run Reports when finished viewing run results.

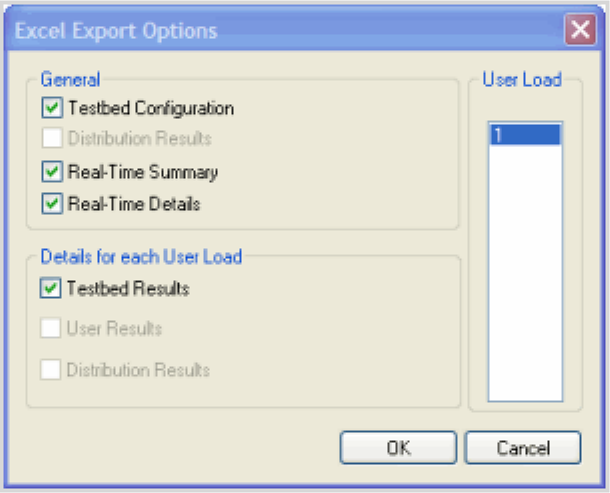
Export Benchmark Factory Test Results to Excel

You can export test results to Microsoft Excel from the Compare Results page and from the Run Reports component.

Note: Microsoft Excel must be installed to perform the following procedure.

To export load test results from Run Reports

1. In Run Reports, click the Excel icon .



Excel Export Options

General

☑ Testbed Configuration

□ Distribution Results

☑ Real-Time Summary

☑ Real-Time Details

Details for each User Load

☑ Testbed Results

□ User Results

□ Distribution Results

User Load

1

2. Modify export options or accept the default settings.
3. Click **OK**. The test results open in Excel.
4. The test results open in Excel.
5. You can then save the report as an Excel file.

To export test results to Excel from the Compare Results page

1. Select a test run and click the Excel icon .
2. Modify export options or accept the default settings.
3. Click **OK**. The test results open in Excel.
4. The test results open in Excel.
5. You can then save the report as an Excel file.

Exporting Test Results to Excel Using the Command Line

Benchmark Factory allows you to export testing results to Excel using the Command line.

To export testing results from Benchmark Factory to Excel using the command line:

1. Select **Start | All Programs | Accessories | Command Prompt**.
2. Navigate to the Benchmark Factory bin directory. For example:

```
CD C:\Program Files\Quest Software\Benchmark Factory\bin

CD C:\Program Files\Quest Software\Benchmark Factory <version
number>\bin
```

3. Enter **BMFRunHistory.exe –x 34 88 108**. Benchmark Factory testing results with Run ids of 34, 88, and 108 export to Excel.

Benchmark Factory Testing Results Terminology

The following provides a list of commonly used Benchmark Factory testing terms.

The following terms can be expressed as:

Average – The average of all recorded values for the statistic over the sampling period.

Minimum – The minimum value the statistic obtained over the sampling period.

Maximum – The maximum value the statistic obtained over the sampling period.

90th Percentile – This is usual associated with a timing statistic. This is the time value where 90 percent of all values recorded for a statistic fell below.

Term	Definition
Bytes	The number of bytes from data received from a SQL statement.
Bytes Per Second	The number of bytes processed per second over the sampling period. This is just the Total Bytes divided by the Sampling period in seconds.
Response Time	The time it takes from when the SQL is sent to the server responds.
Retrieval Time	The time it takes from the server responds to a SQL statement till the last byte of data/results is obtained.
Rows	The number of rows received from a SQL statement.
Rows Per Second (RPS)	The number of rows received per second over the sampling period. Similar to above.
Transactions Per Second (TPS)	The transactions, or SQL statements, processed by the server per second. A transaction in Benchmark Factory can be more than a single SQL statement, such as the TPC-C transaction New Order, this transaction inserts a new order by inserting one record into the new order table and 5 – 7 items in the orderline table.
Transaction Time (sometimes listed as just Time)	The sum of the Response and Retrieval time.

13

BFScripts

About Scripts

Note: All edit field with a yellow background are scriptable.

Benchmark Factory provides a scripting capability as a means of inserting built-in functions for use in scripts. A function is a formula that takes one or more values (arguments), performs an operation, and returns a value that simulates real-world user activity. Functions are used alone or as building blocks in creating complex user activity. Randomized data is important when attempting to simulate real-world user activity because data that is random, prevents a server from using data stored in its cache. Click here for procedures on how to use scripts.

The following provides a list of available scripts:

Category	Script Function
Dates and Numbers	$BFCreditCardExp
	$BFCurrentDate
	$BFCurrentDateTime
	$BFDate
File Access	$BFFileArray
Global Variables	$BFGetGlobalVar
	$BFSetGlobalVar
	$BFSetGlobalVarRtn
Name and Address	$BFAddress
	$BFAddress2
	$BFCity
	$BFCompany
	$BFCountry
	$BFEMail
	$BFFirstName
	$BFFullName
	$BFLastName
	$BFMiddleInitial
	$BFPhone
	$BFState
	$BFZipCode
Numerical Manipulation	$BFFormat
	$BFProd
	$BFSum
Random Numbers	$BFRand
	$BFRandRange
	$BFURand

	$BFURandRange
Random Strings	$BFCreditCard $BFRandList $BFRandMultiList $BFRandStr $BFURandList $BFList
String Manipulation	$BFAsc $BFChr $BFConcat $BFLeft $BFLen $BFLower $BFMid $BFRight $BFTrim $BFTrimLeft $BFTrimRight $BFUpper
Test Info	$BFGetVar $BFMaxNode $BFNode $BFNumberOfIterations $BFProfile $BFRunID $BFSetVar $BFSetVarRtn $BFUserCounter $BFUserID $BFUserLoad $BFNextUserload $BFPrevUserload

BFScript Wizard

Note: Controls with yellow background can be scripted.

Benchmark Factory scripting allows you to randomize the load testing process. Scripting can be utilized by selecting the SQL Statement node and inserting the script in the properties view. A field has scripting capabilities if its right-click menu contains the Script Wizard option.

The Script Wizard is a quick and easy way to use Benchmark Factory scripts. The Script Wizard consists of a script category and a script function. Each script function has a short description included and if applicable, its parameters.

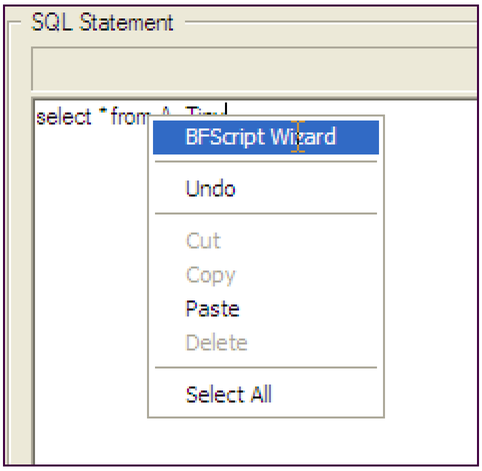
There are two features within Benchmark Factory that assist you when using scripting capabilities:

- Script Wizard
- Script Assist

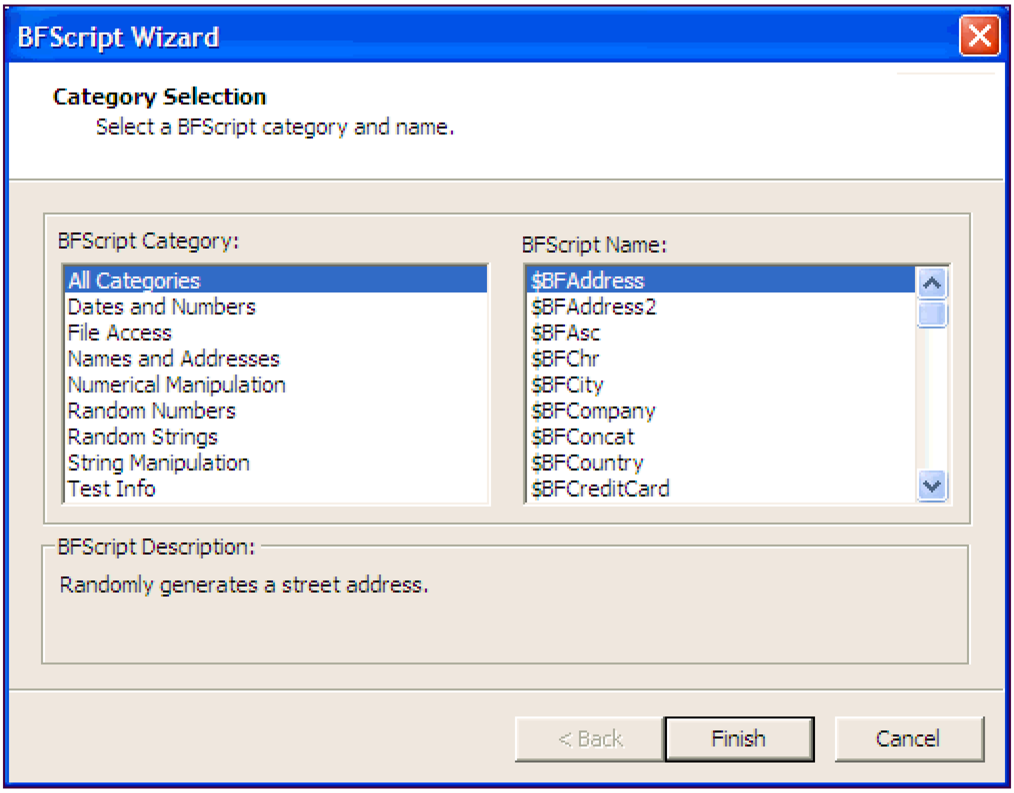
Click here for an overview of $BFScripts.

Script Wizard

1. Right-click the edit control. A drop-down display.



2. Select **BFScript Wizard** from the drop-down. The Script Wizard displays.



3. Select the desired script.

4. Some scripts require further parameters settings. For a complete overview of individual script settings, please refer to Scripts Overview.

   If scripts require parameters:

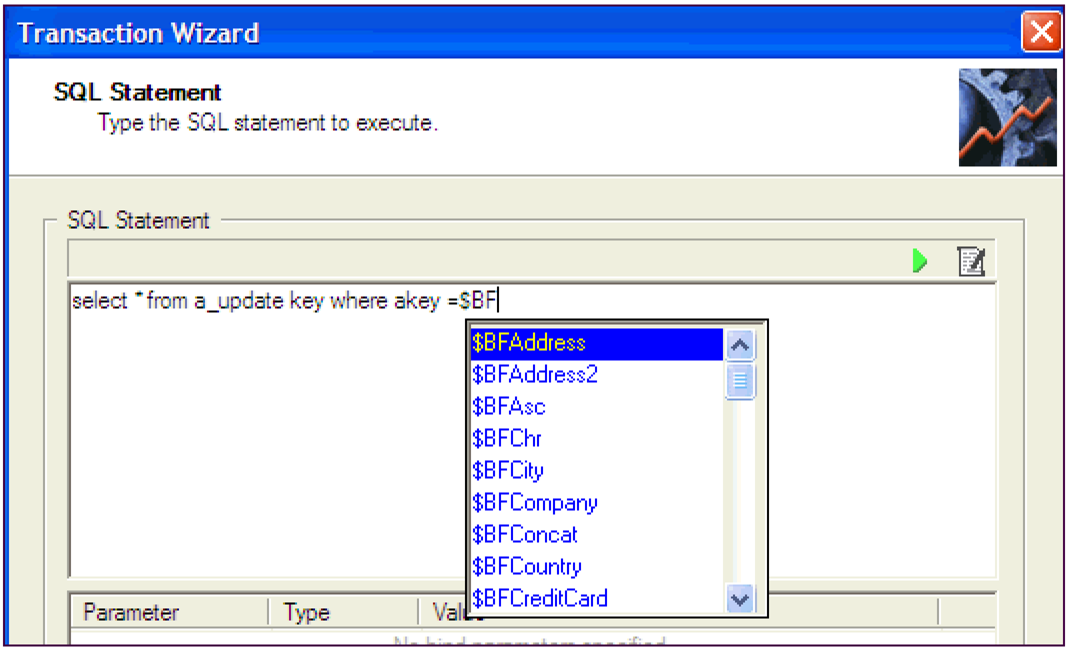
1. Click **Next.**

2. Enter the required parameters.

5. Click **Finish**. The script and parameters display in the Script Wizard.

Script Assist

1. Type in the desired SQL statement. When $BF is typed the Script Assist is activated allowing you to insert the desired script.



```
select * from a_update key where akey =$BF
```

2. Select the desired script.
3. Press **Enter**. The script displays.
4. Type in parameters as required.

Using Scripts in Username and Password Fields

Note: All edit field with a yellow background are scriptable.

Scripts can be used in profiles to login with a variety of users and passwords. If scripts are to be used in a password field, the password needs to be contained entirely in a script such as $BFConcat.

Example: `$BFConcat("USER",$BFUserID(),"X") Will generate: "USER1x" for Virtual User 1; "USER2x" for Virtual User 2; ...`

This example can be useful to log in each virtual user with a unique username and password designated for this use. In this case users should be created in the test system with a username and password containing the virtual user number that will be used.

Dates and Numbers

$BFCreditCardExp

Description:	Gets the global variable.
Parameters:	N/A
Syntax:	`$BFGetGlobalVar('myvar')`

$BFCurrentDate

<table>
<tr><td><b>Description:</b></td><td>Allows you to change the date format used to populate date fields.</td></tr>
<tr><td><b>Parameters:</b></td><td><b>Format string-</b>The mix of static text and variables as needed.
<table>
<tr><th>Variables</th><th>Definition</th></tr>
<tr><td>%a</td><td>Abbreviated weekday name</td></tr>
<tr><td>%A</td><td>Full weekday name</td></tr>
<tr><td>%b</td><td>Abbreviated month name</td></tr>
<tr><td>%B</td><td>Full month name</td></tr>
<tr><td>%c</td><td>Date and time representing appropriate locale</td></tr>
<tr><td>%d</td><td>Day of month as decimal number (01-31)</td></tr>
<tr><td>%H</td><td>Hour in 24-hour format (00-23)</td></tr>
<tr><td>%I</td><td>Hour in 12 hour format (01-12)</td></tr>
<tr><td>%j</td><td>Day of year as decimal number (001-366)</td></tr>
<tr><td>%m</td><td>Month as decimal number (01-12)</td></tr>
<tr><td>%M</td><td>Minute as decimal number (00-59)</td></tr>
<tr><td>%p</td><td>Current locale's A.M./P.M indicator for 12 hour clock</td></tr>
<tr><td>%S</td><td>Second as decimal number (00-59)</td></tr>
<tr><td>%U</td><td>Week of year as decimal number, with Sunday as first day of the week (00-53)</td></tr>
<tr><td>%w</td><td>Weekday as decimal number (0-6; Sunday is 0)</td></tr>
<tr><td>%W</td><td>Week of year as decimal number, with Monday as first day of the week (00-53)</td></tr>
<tr><td>%x</td><td>Date representation for current locale</td></tr>
<tr><td>%X</td><td>Time representation for current locale</td></tr>
<tr><td>%y</td><td>Year without century, as decimal (00-99)</td></tr>
<tr><td>%Y</td><td>Year with century, as decimal number</td></tr>
<tr><td>%z, %Z</td><td>Either the time-zone name or the time zone abbreviation, depending on registry settings; no characters if time zone is unknown</td></tr>
<tr><td>%%</td><td>Percent sign</td></tr>
</table>
</td></tr>
<tr><td><b>Syntax:</b></td><td>$BFCurrentDate (Format String)</td></tr>
<tr><td><b>Example</b>:</td><td>$BFCurrentDate ("%m/%d/%Y) ; returns 03/28/2005</td></tr>
</table>

BFCurrentDateTime

Description:	Allows you to change the date/time format used to populate date fields.
Parameters:	**Format string**-The mix of static text and variables as needed. **Variables** **Definition** %a Abbreviated weekday name %A Full weekday name %b Abbreviated month name %B Full month name %c Date and time representing appropriate locale %d Day of month as decimal number (01-31) %H Hour in 24-hour format (00-23) %I Hour in 12 hour format (01-12) %j Day of year as decimal number (001-366) %m Month as decimal number (01-12) %M Minute as decimal number (00-59) %p Current locale's A.M./P.M indicator for 12 hour clock %S Second as decimal number (00-59) %U Week of year as decimal number, with Sunday as first day of the week (00-53) %w Weekday as decimal number (0-6; Sunday is 0) %W Week of year as decimal number, with Monday as first day of the week (00-53) %x Date representation for current locale %X Time representation for current locale %y Year without century, as decimal (00-99) %Y Year with century, as decimal number %z, %Z Either the time-zone name or the time zone abbreviation, depending on registry settings; no characters if time zone is unknown %% Percent sign
Syntax:	$BFCurrentDateTime (Format String)
Example:	$BFCurrentDateTime ("%m/%d/%Y %H:%M:%S") ; returns 03/28/2005 14:18:52

$BFDate

Description:	Randomly generates a date between a start date and a end date.
Parameters:	**nStart**-The start date. mm/dd/yyyy **nEnd**-The end date. mm/dd/yyyy
Syntax:	$BFDate(nStart,nEnd)
Example:	$BFDate("09/14/2004","09/14/2005") ; returns "11/04/2004"

File Access

$BFFileArray

Description:	Selects an item from a list. Returns a single item from a comma-delimited file. The item returned depends on the mode selected. The syntax of the statement is also slightly different for each mode. Each virtual user gets a different seed value to generate unique sequences. Each agent machine must have a file with the name and path that is specified in the script function. If $BFFileArray is to return strings, the items in the file must be in double-quotes
Parameters:	**File name**-The name and path of the source file. **Retrieval Mode**: CURRENT:Select the item in the specified column of the current row. RANDOM:Select a random item from a given column. SELECTION:Select an item based on column and row. SEQUENTIAL:Select each item sequentially. UNIQUE:Select a non-repeating item from a given column. **nColumn**-Which column of values to select. If none is given, a value of 1 is assumed. nRow-Which row to select for the item. Note: This value is only needed for the SELECTION mode.
Syntax:	$BFFileArray`(FileName, Retrieval Mode[,nColumn][, nRow])`

Example:	If myfile.dat consists of:
	"George Washington", "Martha"
	"Franklin Roosevelt","Eleanor"
	"Gerald Ford","Betty"
	`$BFFileArray( "c:\path\myfile.dat", SELECTION, 2, 3)`
	Returns **Betty**.
	`$BFFileArray( "c:\path\myfile.dat", RANDOM,2)`
	Returns either **Martha**, **Eleanor**, or **Betty**. Will return **Martha**, **Eleanor**, or **Betty** the next time it is run.
	`$BFFileArray( "c:\path\myfile.dat", UNIQUE,2)`
	Returns either **Martha**, **Eleanor**, or **Betty**. Will return **Martha** or **Betty** the next time it is run if it returned Eleanor the first time.
	`$BFFileArray( "c:\path\myfile.dat", SEQUENTIAL, 1)`
	Returns George Washington then Franklin Roosevelt then Gerald Ford.
	`$BFFileArray( "c:\path\myfile.dat", CURRENT, 2)`
	Returns Betty if the previous script function returned Gerald Ford.

Global Variable

$BFGetGlobalVar

Description:	Generates the Get Global Variable.
Parameters:	N/A
Syntax:	`$BFGetGlobalVar('myvar')`

$BFSetGlobalVar

Description:	Generates the Set Global Variable
Parameters:	N/A
Syntax:	`$BFSetGlobalVar('myvar', 'my value')`

$BFSetGlobalVarRtn

Description:	Generates the Set Global Variable Return.
Parameters:	N/A

Syntax:	`$BFSetGlobalVarRtn('myvar', 'my value')`

Name and Address

$BFAddress

Description:	Returns a randomly generated street address string.
Parameters:	N/A
Syntax:	`BFAddress()`
Example:	`$BFAddress() ; returns "904 Oak Lane"`

$BFAddress2

Description:	Returns a second randomly generated street address string containing an apartment number, suite number, or villa.
Parameters:	N/A
Syntax:	$BFAddress2()
Example:	`$BFAddress2() ; returns  "Apt 5442"`

$BFCity

Description:	Generates a random city name.

Parameters:	N/A
Syntax:	`$BFCity()`
Example:	`$BFCity() ; returns "Trend Blue of Asia"`

$BFCompany

Description:	Returns a random company name
Parameters:	N/A
Syntax:	`$BFCompany()`
Example:	`$BFCompany(); returns "Quest Software"`

$BFCountry

Description:	Returns a randomly generated country string.
Parameters:	N/A
Syntax:	`$BFCountry()`
Example:	$BFCounty() ; returns "Canada"

$BFEmail

Description:	Returns a random email address string.
Parameters:	N/A

Syntax:	$BFEmail()
Example:	$BFEmail() ; returns "Rita_Smyth@TrendBlue.net"

$BFFirstName

Description:	Returns a random first name string.
Parameters:	N/A
Syntax:	`$BFFirstName()`
Example:	`$BFFirstName() ; returns: "Hary D. Jones"`

$BFFullName

Description:	Returns a random full name string.
Parameters:	N/A
Syntax:	$BFFullName()
Example:	$BFFullName() ; returns "Harry Jones"

$BFLastName

Description:	Returns a random last name string.
Parameters:	N/A
Syntax:	`$BFLastName()`

Example:	`$BFLastName() ; returns "Jones"`

$BFMiddleInitial

Description:	Returns a random middle initial character.
Parameters:	N/A
Syntax:	`$BFMiddleInitial()`
Example:	`$BFMiddleInitial() ; returns "H"`

$BFPhone

Description:	Returns a randomly generated telephone string.
Parameters:	N/A
Syntax:	`$BFPhone()`
Example:	`$BFPhone() ; returns  "(348)889-6599"`

$BFState

Description:	Returns a randomly generated state string.
Parameters:	N/A
Syntax:	`$BFState()`

Example:	`$BFState() ; returns "CA"`

$BFZipCode

Description:	Returns a randomly generated zip code string.
Parameters:	N/A
Syntax:	$BFZipCode()
Example:	$BFZipCode() ; returns "52076"

Numerical Manipulation

$BFFormat

Description:	Formats a series of up to 16 numbers. If the amount of numbers in the series is greater than 16, the Maximum Parameters Exceeded message displays.
Parameters:	**Format string** -A %d for each number in the series. **n1**-The first number in the series. **n2**-The second number in the series. **n16**-The last number in the series.
Syntax:	`$BFFormat(Format string, n1, n2, …, n16)`
Example:	$BFFormat("%d is a crazy number", $BFRandRange(10,200)) ; returns "34 is a crazy number"

$BFProd

Description:	Returns the product of a series of floating-point numbers.
Parameters:	**f1**-A floating-point number. **f2**- A second floating-point number. **fn**...-The last floating-point number.
Syntax:	`$BFProd(f1, f2, ... fn)`
Example:	`$BFProd(5, 2.5, .01) returns ; "0.125"`

$BFSum

Description:	Returns the summation of a series of numbers.
Parameters:	**f1**-The first number to be summed. **f2**-The second number to be summed. **fN**-The last number to be summed.
Syntax:	`$BFSum(f1, f2, ..., fN)`
Example:	`$BFSum(10, 20, 30) ; returns "60"`

Random Numbers

$BFRand

Description:	Returns a random integer between 0 and nMax. Each virtual user gets a different seed value to generate the same unique sequences for each run.

Parameters:	**nMax**-The maximum integer to be returned by the function.
Syntax:	$BFRand(nMax)
Example:	$BFRand(100) ; returns "45"

$BFRandRange

Description:	Returns a random integer ranging between nMin and nMax inclusive. Each virtual user gets a different seed value to generate sequences.
Parameters:	**nMin**-The minimum integer in a range. **nMax**-The maximum integer in a range.
Syntax:	`$BFRandRange(nMin,nMax)`
Example:	$BFRandRange(200, 210) ; returns "202"

$BFURand

Description:	Returns a **unique** (non-repeating) random integer ranging between 1 and nMax. Each virtual user gets a different seed value to generate unique sequences for each run.
Parameters:	**nMax**-The maximum integer to be returned.
Syntax:	`$BFURand(nMax)`
Example:	$BFURand(100) ; `returns 78 the first time this function executes, and 50, and 19 for subsequent executions of this function.`

$BFURandRange

Description:	Returns **unique** integers ranging between the value of nMin and the value of nMax inclusive. Each virtual user gets a different seed value to generate unique sequences for each run.
Parameters:	**nMin**-The minimum range integer to return. **nMax**-The maximum range integer to return.
Syntax:	`$BFURandRange(nMin,nMax)`
Example:	$BFRandRange(1,100) ; returns 100 the first time this function executes and 95, and 85 for subsequent executions of this function.

Random String

$BFCreditCard

Description:	Returns a random credit card number string.
Parameters:	N/A
Syntax:	`$BFCreditCard()`
Example:	$BFCreditCard() ; returns "7970 2384 8889 6599"

$BFRandList

Description:	Returns a string randomly selected from the list of items. If no weight is specified, a weight of 1 is assumed. Each virtual user gets a different seed value to generate unique sequences.
Parameters:	**string1**-The first string in a list to return.

	nWeight1-Positive integer indicating the relative weight of the first string. **string2**-The second string in a list to return. **nWeight2**-Positive integer indicating the relative weight of the second string. **stringN-**The last string in a list to return. **nWeightN**-Positive integer indicating the relative weight of the last string.
Syntax:	`$BFRandList(string1[:nWeight1], string2[:nWeight2], …,stringN [:nWeightN])`
Example:	$BFRandList("Red", "White":2, "Blue":5) The value White will be returned 2 times more often than the value Red. The value Blue will be returned 5 times more often than the value Red.

$BFRandMultiList

Description:	Randomly selects multiple strings based on probabilities from a list. If Weight is omitted, a value of 100 is assumed. The probability that any string is include in the returned value is determined by the value of nWeight. Each string included in the return value is separated by a comma.
Parameters:	**string1**-The first string in a list to return. **nWeight1**-Positive integer indicating the relative weight of the first string. Probability should be expressed as a value between 0 and 100. **string2**-The second string in a list to return. **nWeight2**-Positive integer indicating the relative weight of the second string. Probability should be expressed as a value between 0 and 100. **stringN-**The last string in a list to return. **nWeightN-**Positive integer indicating the relative weight of the last string. Probability should be expressed as a value between 0 and 100.
Syntax:	`$BFRandMultiList(string1[:nWeight1], string2[:nWeight 2], …, stringN"[: nWeightN])`
Example:	$BFRandMultiList("Red", "White":50, "Blue":25) ; returns a value that always contains Red, contained White 50% of the transactions, and Blue 25% of the transactions.

$BFRandStr

Description:	Returns a random string determined by a mode and having a length n.
Parameters:	**Length n**-The length of a string. **Mode constant**-One of the following: **ALPHA**-Returns only random letters. **ALPHA_LOWER**-Returns random lowercase letters. **ALPHA_NUM**-Returns random letters and numbers. **ALPHA_UPPER**-Returns random uppercase letters. **ANY**-Returns any random ASCII characters. **NUM**-Returns only numbers. **RANGE**-Returns ASCII characters in the range specified.
Syntax:	$BFRandStr(n,[constant],[1..255],[1..255])
Example:	$BFRandStr(10) ; Returns 10 random characters. $BFRandStr(1,RANGE,65,67) ; Returns either A, B, or C.

$BFURandList

Description:	Randomly select unique items from a list.
Parameters:	**string1**-The first string to return from a list. **string2**-The second string to return from a list. **stringN**-The last string to return from a list.
Syntax:	$BFURandList(string1,string2,… stringN)
Example:	$BFURandList("Red", "White", "Blue", "Yellow") ; returns Blue the first time this function executes then Yellow, Red, White, and Blue for subsequent

	executions of this function. Notice that Blue did not repeat until all other colors returned.

$BFList

Description:	Returns an item from a list. The item returned depends on the mode selected.
Parameters:	**Retrieval Mode**: RANDOM: Select a random item from the list. SEQUENTIAL: Select each item sequentially. UNIQUE: Select a non-repeating item from the list. **string1**-The first string to return from a list. **string2**-The second string to return from a list. **stringN**-The last string to return from a list.
Syntax:	$BFList(Retrieval Mode, string1,string2,… stringN)
Example:	$BFList (Sequential "1", "2", "3", "4") Returns 1 2 etc. $BFList (Random 1, 2, 3, 4) Returns 2 3 3 1 2 4 etc. $BFList (Unique, "1", "2", "3","4") Returns 2 4 3 1 2 etc.

String Manipulation

$BFAsc

Description:	Returns the ANSI value of the first character of a string.
Parameters:	**String**-Characters enclosed in quotation marks.
Syntax:	$BFAsc(string)
Example:	$BFAsc("George") ; Returns "71"

$BFChr

Description:	Returns the character associated with the specified ANSI code.
Parameters:	**n**-An integer representing an ANSI code.
Syntax:	$BFChr(n)
Example:	$BFChr(68) ; returns "D"

$BFConcat

Description:	Returns a string containing two or more strings.
Parameters:	**string1**-The first string to return. **string2**-The second string to return. **stringN**-The last string to return.
Syntax:	$BFConcat("string1", "string2", …, stringN)
Example:	$BFConcat ("The connection interval ", "determines how often an ", "agent attempts to connect to the Benchmark Factory Console.") ; returns "The connection interval determines how often an agent attempts to connect to the Benchmark Factory Console"

$BFLeft

Description:	Returns the first n character of a string.
Parameters:	**String**-Characters enclosed in quotation marks. **nLength**-The length of the string to return.
Syntax:	`$BFLeft(string,length)`
Example:	`$BFLeft("Benchmark Factory", 5) ; returns  "Bench"`

$BFLen

Description:	Returns the number of characters in a string.
Parameters:	**String**-Characters enclosed in quotation marks.
Syntax:	$BFLen(string)
Example:	$BFLen("Benchmark Factory") ; returns "17"

$BFLower

Description:	Returns a string after converting uppercase characters to lowercase characters.
Parameters:	**String**-Characters enclosed in quotation marks.
Syntax:	$BFLowerstring)
Example:	$BFLower("SAMPLING") ; returns "sampling"

$BFMid

Description:	Extracts a substring from a string.
Parameters:	**String**-Characters enclosed in a quotation marks. **nStart**-The starting character position. **nLength-**Number of characters to return.
Syntax:	$BFMid(string, nStart,nLength)
Example:	$BFMid("sunshine coast", 3, 5) ; returns "shine"

$BFRight

Description:	Returns the last n character of a string.
Parameters:	**String**-Characters enclosed in quotation marks. **nLength**-The number of characters to return.
Syntax:	$BFRight(string,nLength)
Example:	$BFRight("sunshine coast", 3) ; returns "ast"

$BFTrim

Description:	Returns a string void of leading and trailing spaces.
Parameters:	**String**-Characters enclosed in quotation marks.
Syntax:	$BFTrim(string)
Example:	$BFTrim(" happy days are here to stay. ") ; returns "happy days are here to stay"

$BFTrimLeft

Description:	Returns a string void of leading spaces.
Parameters:	**String**-Characters enclosed in quotation marks.
Syntax:	$BFTrimLeft(string)
Example:	$BFTrimLeft(" hockey is great ") ; returns " hockey is great "

$BFTrimRight

Description:	Removes trailing spaces from a string.
Parameters:	**nString**-Characters enclosed in quotation marks.
Syntax:	$BFTrimRight(string)
Example:	$BFTrimRight(" removes trailing spaces from a string. ") ; returns " removes trailing spaces from string. "

$BFUpper

Description:	Returns a string after converting lowercase characters to uppercase characters.
Parameters:	**nString value**-Characters enclosed in quotation marks.
Syntax:	`$BFUpper(string)`
Example:	$BFUpper("sampling") ; returns "SAMPLING"

Test Info

$BFGetVar

Description:	Retrieves a previously stored value using $BFSetVar. Allows a value to be passed from one transaction to another in conjunction with $BFSetVar, or when value is used multiple times within a transaction. Each virtual user has its own variable space, so values are not shared between them.
Parameters:	**VarName**-An alphanumeric identifier of the value stored.
Syntax:	`$BFGetVar("VarName")`
Example:	$BFSetVar("Totalrow", "2") $BFGetVar("Totalrow") ; if $BFSetVar sets the variable "Totalrow" to 2, $BFGetVar("Totalrow") returns the number 2.

$BFMaxNode

Description:	Returns the total number of nodes for all users. This function is intended only for Oracle clustering.
Parameters:	N/A
Syntax:	$BFMaxNode()
Example:	$BFMaxNode() ; returns: 2. This is the same value seen in the Number of Nodes field in Clustering tab for an Oracle profile.

$BFNode

Description:	Returns the node number of the current user. This function is intended only for Oracle clustering.
Parameters:	N/A
Syntax:	$BFNode()
Example:	$BFNode() ; returns "1"

$BFNumberOfIterations

Description:	Returns the current number of iterations of a test.
Parameters:	N/A
Syntax:	$BFNumberOfIterations()
Example:	$BFNumberOfIterations() ; returns "1"

$BFProfile

Description:	Returns driver specific information, such as database name.
Parameters:	**Profile (constant)**-The following provides a list of database type constants: **MSSQL (native), SAP:** • SERVERNAME • DATABASE • USERNAME **ORACLE**: • NETSERVICENAME • USERNAME **ODBC, MYSQL (ODBC), MSSQL (ODBC):** • DATASOURCENAME • USERNAME **DB2:** • DATABASEALIAS • USERNAME
Syntax:	$BFProfile(constant)
Example:	$BFProfile(USERNAME) ; returns 'sa'

$BFRunID

Description:	Returns the run ID of the current test.
Parameters:	N/A
Syntax:	`$BFRunID()`
Example:	$BFRunID() ; returns "186"

$BFSetVar

Description:	Stores a value for later use by $BFGetVar. Used to store a value to be reused within its own transaction, or any transaction in a given user scenario. Each virtual user gets its own variable space, so values are not shared between them. Typically, $BFSetVar is placed at the beginning of a dynamic statement, as scripts are evaluated from left to right.
Parameters:	**Variable Name (VarName)**-An alphanumeric identifier of the value stored. **Text to Store (Value)**-A string. The value to be stored for later retrieval.
Syntax:	Syntax:$BFSetVar("VarName", "Value")
Example:	$BFSetVar("Totalrow", "2") ; $BFSetVar sets the variable "Totalrow" to 2

$BFSetVarRtn

Description:	Stores and returns a value to be reused within its own transaction, or any transaction in a given user scenario. Each virtual user gets its own variable space, so values are not shared between them. Typically, $BFSetVarRtn is placed at the beginning of a dynamic statement, as scripts are evaluated from left to right.
Parameters:	**Variable Name (VarKey)**-A string to store the a value. **Text to Store (Value)**-A string value to be stored for later retrieval.
Syntax:	$BFSetVarRtn("Varkey","Value")
Example:	$BFSetVarRtn("Totalrows", "2") ; returns "2"

$BFUserCounter

Description:	Returns the user counter.
Parameters:	N/A
Syntax:	`$BFUserCounter()`
Example:	`$BFUserCounter() ; returns "1"`

$BFUserID

Description:	Returns the current virtual user ID.
Parameters:	N/A
Syntax:	$BFUserID()
Example:	$BFUserID() ; returns "1"

$BFUserLoad

Description:	Returns the current user load for the test running.
Parameters:	N/A
Syntax:	$BFUserload()
Example:	$BFUserLoad() ; returns "4"

$BFNextUserload

Description:	Returns the next user load for the load scenario running.
Parameters:	N/A
Syntax:	$BFNextUserload()
Example:	$BFNextUserload() When running with userloads 1, 4, 6, 10 this will return "6" when running at userload 4.

$BFPrevUserload

Description:	Returns the previous user load for the load scenario running.
Parameters:	N/A
Syntax:	$BFPrevUserload()
Example:	$BFPrevUserload() When running with userloads 1, 4, 6, 10 this will return "4" when running at userload 6.

14

Repository

Repository Manager

Note: If you create a new Benchmark Factory 5.5 (or later) repository, earlier versions of Benchmark Factory will not work against this repository.

The Repository is a database where all of the test results are stored. Benchmark Factory inserts test results into the repository and provides an easy way to access the data. By default, the Repository is a SQLite database that resides on the same machine as Benchmark Factory. The Repository can reside on another database server if required.

Note: By default in Benchmark Factory 7.1.1 or earlier, a MySQL database is created and used as the Repository, unless you selected the SQLite option during installation. In Benchmark Factory 7.2 or later, by default a SQLite database is created and used as the Repository.

To change the database, select the Data Source Name of the ODBC connection for the new database. To migrate data from one database to another, click **Data Migration** to open the Data Migration Wizard.

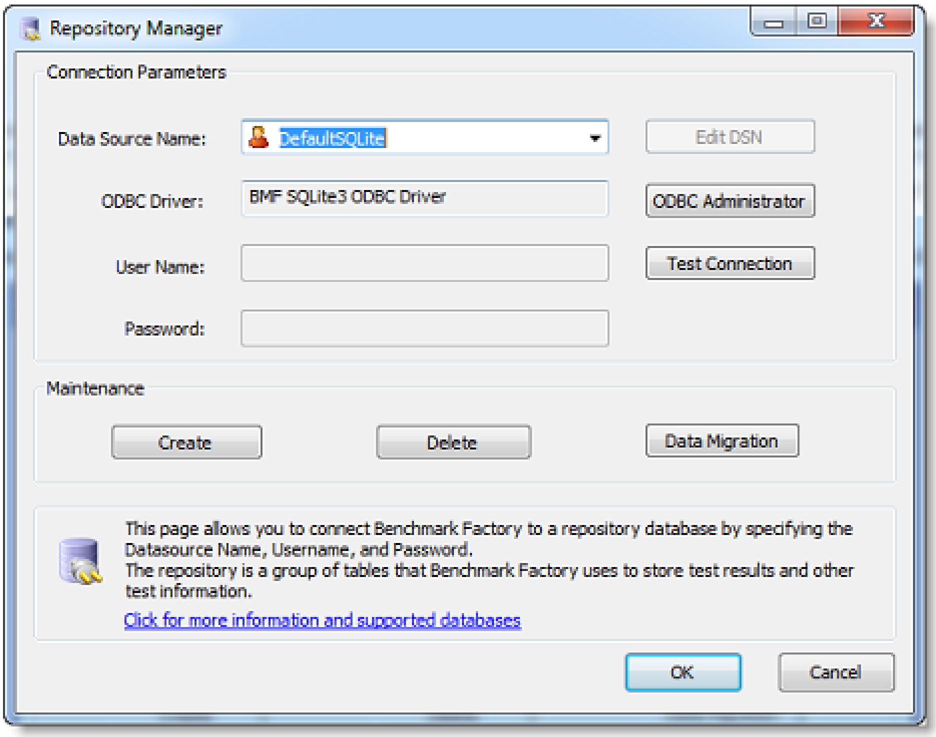
Note: If the database structure does not exist on the selected database, a prompt to create the structure will appear when **OK** is clicked.

The supported database types are:

- MySQL
- SQL Server®
- SQLite
- SAP SQL Anywhere® (Adaptive Server® Anywhere)
- Oracle®

To open the Repository Manager

1. To open the Repository Manager from the application, select **Tools | Repository Manager**. Benchmark Factory prompts you to close the application. This is necessary before opening the Repository Manager.
2. The Repository Manager opens. Modify connection parameters or perform maintenance functions.



Data Source Name: DefaultSQLite

ODBC Driver: BMF SQLite3 ODBC Driver

User Name:

Password:

This page allows you to connect Benchmark Factory to a repository database by specifying the Datasource Name, Username, and Password.
The repository is a group of tables that Benchmark Factory uses to store test results and other test information.

Review the following for additional information:

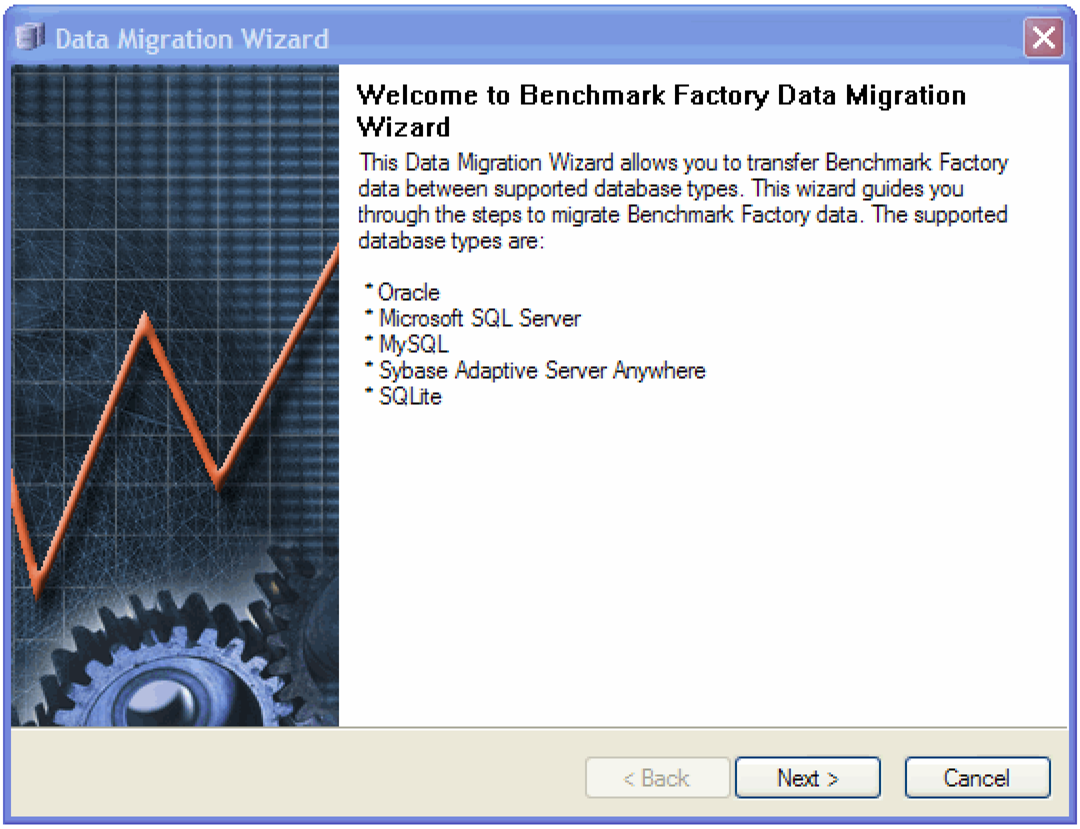
Connection Parameters	
Data Source Name	Data Source name of the ODBC connection used to connect to the repository database.
ODBC Driver	Current ODBC driver
User Name	The User Name used to log into the selected database.
Password	The Password associated with the user name used to log into the database.
Edit DSN	Displays the ODBC connection information dialog for the selected data source.
ODBC Administrator	Displays the ODBC Data Source Administrator dialog. Use this to add and edit ODBC connections.
Test Connection	Tests the connection of the currently selected ODBC Data Source.
Maintenance	
Create	Creates the repository objects on the selected database.
Delete	Deletes the repository objects on the selected database. ℹ **Warning**: This will delete all test results stored in the Repository.
Data Migration	Launches the Data Migration Wizard.

Data Repository Migration Wizard

The Data Repository Migration Wizard allows you to migrate the Benchmark Factory testing repository from one database type to another. For example, from the default SQLite database to a Microsoft SQL Server database.

To use the Data Repository Migration Wizard

1. To open the wizard from the Benchmark Factory application, select **Tools | Repository Data Migration**.
2. When the Data Migration Wizard opens, click **Next**.



Welcome to Benchmark Factory Data Migration Wizard

This Data Migration Wizard allows you to transfer Benchmark Factory data between supported database types. This wizard guides you through the steps to migrate Benchmark Factory data. The supported database types are:

- Oracle
- Microsoft SQL Server
- MySQL
- Sybase Adaptive Server Anywhere
- SQLite

3. In the Choose a Data Source page, select the data source name for the database from which you want to migrate Benchmark Factory data.
   - To migrate data from the default SQLite database installed and used with any *new* installation of Benchmark Factory 7.2 or later, select **Default SQLite**.

     Note: The option to use a default SQLite database was also available in Benchmark Factory 7.1.1 or earlier.
4. Enter the user name and password, if necessary (for example, if migrating from a non-default database). Click **Next**.
5. In the Choose a Destination page, select the data source name of the database to which you want to migrate the data.
6. Click **Next**. The Data Migration Wizard completion dialog displays.
7. Click **Finish.**

15

Troubleshooting

Support Bundle

The Benchmark Factory Support Bundle allows you to zip up a .bfp file and send it to support for review. The bundle contains the following information:

- BMF version number
- Settings
- License information (just send the key files)
- Error Logs and Result Logs from the associated directories
- Files located in the data directory. This will be not only the xml files for imported users scenarios, but also dump files.
- Script files XML configuration files
- A file that contains hardware information about the system running on it as well as the output of the agents configurations from the repository
- Information about versions of all loaded .dlls

To create a Benchmark Factory support bundle

1. Select **Help | Support Bundle** to open the Support Bundle dialog.
    - To create a support bundle for a single job, select the job in the Jobs View pane before opening the Support Bundle dialog.
2. In the Support Bundle dialog, select the modules you want to send.
3. If creating a support bundle for a single job, select **Just for selected job**.
4. Then do one of the following:
    - To save the bundle, click the **Save** icon.
    - To email the bundle to Support, click **Email Quest Software Support**.
    - To contact Support via the Support Portal, click **Quest Support**.

    The bundle is created and saved in the following location:

    > C:\Documents and Settings\<*user name*>\My Documents\Benchmark Factory\<*version number*>\BMFSupportBundle.zip.

Agent Connection

If the agent is having a problem connecting to the Benchmark Factory console, please check the following:

- Verify that the console is running.
- Verify that the agent is pointed to the console machine. In the Agent dialog, go to **Options | Settings** and check the Machine Name/IP and the Console Port fields . See Configure Agent Machine on page 145 for more information.
- Verify TCP connectivity between the Agent and Console. If there is a firewall in place, allow for communication on port 4568 (or the specified console port). See Configure Firewall for Remote Agent Install/Start-Up on page 252 for more information.

Troubleshooting Standard Benchmark Tests

Review the following to help troubleshoot issues that might occur during standard benchmark testing.

Issue	Cause/Solution
"Bad packet" error when testing against Oracle 12c on Linux	When testing against an Oracle 12c database in a Linux environment, if you encounter a "bad packet" error when loading benchmark objects, you might attempt the following workaround. **Workaround:** You can reduce the number of rows per commit by adding a key to the Registry. 1. In the Registry, navigate to HKEY_CURRENT_USER\Software\Quest Software\Benchmark Factory for Databases\Benchmark Factory Console\Settings. 2. Add the following DWORD key: RowsPerCommit. 3. Specify a value for the number of rows per commit, for example 25. 4. Then navigate to HKEY_CURRENT_USER\Software\Quest Software\Benchmark Factory for Databases\Benchmark Factory Agent\Settings. 5. Add the same DWORD key and the same value. ! **Caution: Editing your Registry incorrectly can cause serious, system-wide problems. It is advised that you back up your registry before modifying it.**

Use Benchmark Factory with SQL Server 2005 Client

For connections to Microsoft SQL Server 2005 (or later), use the SQL Server native connection or ODBC connection. See https://msdn.microsoft.com/en-us/library/cc280356(v=sql.120).aspx for more information about SQL Server Native Client support.

MySQL Initialization Settings

The MySQL repository database can take up a large amount of space on a system. The my.ini file allows you to edit the initialization settings and adjust the size of the MySQL repository used by Benchmark Factory. For additional information on setting MySQL database parameters, please visit MySQL.com.

ℹ **Note:** By default in Benchmark Factory 7.1.1 or earlier, a MySQL database is created and used as the Repository, unless you selected the SQLite option during installation. In Benchmark Factory 7.2 or later, by default a SQLite database is created and used as the Repository.

16

Licensing

Licensing and Virtual User Packs

The Benchmark Factory Licensing dialog allows you to enter a new license key, modify a license, or modify the virtual user pack.

The active license or licenses in Benchmark Factory determine the available features, including the ability to connect to certain types of databases. Benchmark Factory supports multiple licenses. This can enable multiple license-dependent features.

Note: If you delete an active Trial key from the Licensing dialog, any features enabled by the Trial key and not included in one of the remaining active licenses will be disabled.

To access the licensing dialog

1. Click **Help** | **Licensing**.

   From this dialog you can:

   - Enter a New License Key
   - Modify the Benchmark Factory License
   - Modify the Benchmark Factory Virtual User Pack
   - Upgrade Earlier-Version License Key

Enter a New License Key

To enter a new license key

1. In the Licensing dialog, click the blank tab.
2. Enter your Authorization key and the site message you received with your key.

Modify the Benchmark Factory License

If required, you can modify your Benchmark Factory license key.

To modify the license key

1. Click the tab for the product license key you want to modify.
2. Click **Modify**.
3. Update the Authorization Key and Site Message.

Modify the Benchmark Factory Virtual User Pack

Benchmark Factory comes with 100 virtual users. Benchmark Factory Freeware comes with 20 virtual users. To add up to an unlimited number of virtual users, enter a virtual user key.

To modify the Virtual User Pack

1. Click the Virtual User Pack tab.
    a. To send an email to Quest Software to extend a trial, click **Extend Trial**.
    b. To visit the Quest Software Web site and purchase a license, click **Purchase License**.
    c. To modify the current license, click **Modify**.
    d. Enter the authorization key and site message.

Upgrade Earlier-Version License Key

License keys compatible with Benchmark Factory 5.8.1 (or earlier) are no longer supported.

To upgrade your license key

1. Go to **Help | Licensing** to find your license *number* in your current version of Benchmark Factory. If not explicitly listed, it may be included as part of the Site Message associated with your key.

    Example: ACCOUNTNAME-nnn-nnn-nnn

2. Then go to the License Key Upgrade page: http://license.quest.com/upgrade.
3. Enter your e-mail address and your existing license *number*, and then follow the prompts.

If you need help finding your license number or an upgrade key, please contact the License Administration team: https://support.quest.com/licensing-assistance.

Add Virtual Users

Benchmark Factory comes with 100 virtual users by default. Benchmark Factory Freeware comes with 20 virtual users. Virtual users simulate the load against the system-under-test. Each virtual user is a separate thread, acting independently of the other virtual users, with its own connection to the system-under-test. Each virtual user tracks its own statistics that includes transaction times and the number of times a transaction executes.

Benchmark Factory can scale up to an unlimited number of concurrent load users. Additional virtual users allow you to:

- Accurately determine system throughput and capacity for database systems
- Simulate thousands of concurrent users with a minimal amount of hardware
- Find applications that do not scale well with an increase in the number of users
- Find breaking points, weak links, or bottlenecks of a system
- Quantify application or server performance with realistic workloads

To request additional concurrent load users, please contact your Quest Software representative, or visit the Benchmark Factory Web site.

17

The Benchmark Factory REST API

Benchmark Factory provides a REST API that allows you to access the functionality of Benchmark Factory, but without the need to interact with the Benchmark Factory graphic user interface. A REST API is an application program interface (API) that uses HTTP requests to GET, POST, PUT, and DELETE data. Through the REST API, you can use a script, command-line tool, or custom application to automate your load testing tasks using Benchmark Factory.

You can use the Benchmark Factory REST API with the Benchmark Factory console application or BMFServer.exe. BMFServer.exe is a non-UI application included in the Benchmark Factory installation. See BMFServer.exe for more information.

Why Integrate Benchmark Factory

Integrating Benchmark Factory into your continuous testing / continuous improvement process allows you to test the scalability of changes before committing those changes to your production environment. Different Benchmark Factory tests can be used to validate scalability and performance.

- Use **Industry Standard Benchmark tests** to test for scalability
- Use **Capture and Replay tests** to verify application performance

Base URI Format

To interface with Benchmark Factory using the REST API, use the following URI format:

http://{server}:{port}/api/{resource}

Where {server} is the Benchmark Factory host server, {port} is the port number, and {resource} is the name of the resource.

Example: http://localhost:30100/api/jobs

Console and BMFServer.exe Port Numbers

Use the applicable REST API port number for the Benchmark Factory application you are using.

- Benchmark Factory Console: port 30100
- BMFServer.exe: port 30110

HTTP Request Methods

GET - Retrieves a list of resources

POST - Creates a new resource based on the data provided in the request body

PUT - Updates the resource to the value/values provided in the data supplied in the HTTP data area

DELETE - Deletes the resource

Start Benchmark Factory

Benchmark Factory must be running to use the REST API. Start the Benchmark Factory console or BMFServer.exe prior to sending requests via the REST API.

Note: You cannot run the Benchmark Factory console and BMFServer.exe on the same machine at the same time.

Request and Response Format

Both XML and JSON are supported request and response formats. Specify the response format in the Accept header of the request. Indicate the request format in the Content-Type header of the request.

New Resource Name/Index

When you create a resource, such as a job or connection, the new resource is assigned a zero-based index. This number can be used to identify the resource. However, the index number attached to the resource will change as users remove and add resources. For best results, define a name for the resource and then use the resource name in your requests.

Response Header Content

If a resource (such as a connection or a job) is created, the response header usually contains the URL for the newly-created resource.

Response Codes

These are the expected response codes returned. In addition, some other status codes may be returned if either an internal error occurs or there are other issues.

Code	Description
200	Success
201	Success. Resource created.
204	Success. No content returned.
400	The request failed.
404	The specified resource was not found.
405	The method is not supported for the specified resource.

Job Status Codes

Each job state or status has a code. These are the codes returned when you get the job status. Use these codes to change a job’s status (e.g., start a job, stop a job).

Code	Status	Action when included in a request
1	Complete	Stops a running job.
2	Running	Runs a job, if possible.

Code	Status	Action when included in a request
3	Ready	Readies a job to run on schedule. Stops a running job.
4	Hold	Prevents a job from running. Stops a running job.
5	Stop	Stops a running job.

Resource Structure and URI Hierarchy

The following represents the object structure for the resources in the Benchmark Factory REST API.

Connections

Connections	
	Connection

Jobs

Jobs					
	Job				
		Connection			
		Agents			
			Agent		
		Tests			
			Test		
				Transactions	
					Transaction
		Schedule			

TestRuns

TestRuns							
	TestRun						
		Connection					
		Test					
		UserLoadResults					
			UserLoadResult				
				TransactionResults			
					TransactionResult		
						SubTransactionResults	
							SubTransactionResult
		NodeResults					
			NodeResult				
		TestFiles					
			TestFile				
		Agents					
			Agent				
				DriverInfos			
					DriverInfo		

Settings

Settings					
	Benchmark				
		AS3AP			
		ObjectDataRetention			
		Replication			
			TableStructure		
				Columns	
		Scalable Hardware			Column
		TPC-B			
		TPC-C			
		TPC-D			
		TPC-E			
		TPC-H			
	ErrorHandling				
		Threshold			
	ExecuteFile				
	General				
		Directories			
			Directory		
		Retrieval			
		TCPIP			
	Agent				
		UserAgents			
	Latency				
		InterarrivalTime			
		KeyingTime			
		ThinkTime			
	Oracle				
		AWRADDM			
		Statspack			
	Repository				
	Statistics				
	Timing				
		UserStartup			
	Userload				

REST API Resources

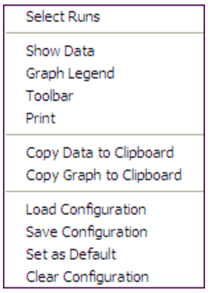
For a list of Benchmark Factory REST API resources, including resource descriptions and examples, please see http://www.toadworld.com/products/benchmark-factory/w/wiki/12225.bmf-rest-api.

18

Appendix

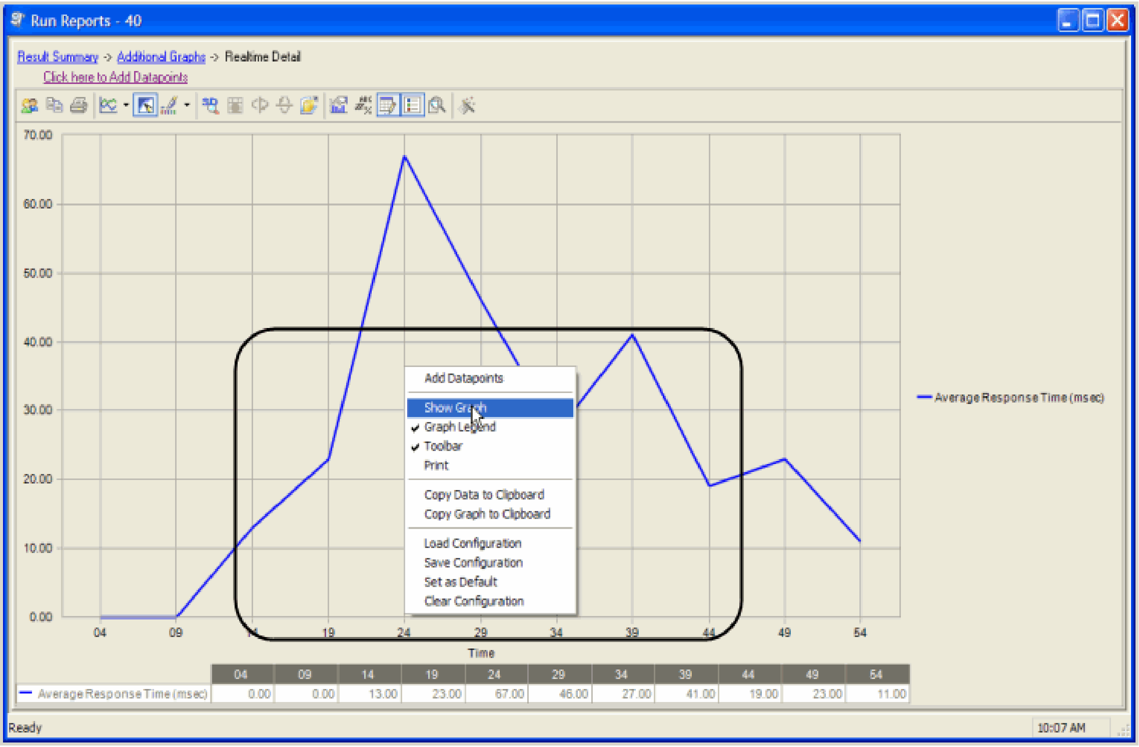
Change Graph Views

Right-clicking on a Benchmark Factory graph displays a drop-down that allows you to customize graph settings. Whether you are viewing a real-time graph or one in Benchmark Factory Run Reports, Benchmark Factory uses the same graphing tool. In most cases you will find that the graphs are presented in the most meaningful form (Line Graphs or Bar Graphs based on the data being presented.)

	Displays Benchmark Factory Run Reports Shows in a data form or graph form Displays the Graph Legend Displays the graph toolbar Prints the graph Copies Data to Clipboard Copies Graph to Clipboard Loads a saved graph configuration Saves a graph configuration Sets the current configuration as default Clears the current chart configuration

Show Data/Show Graph

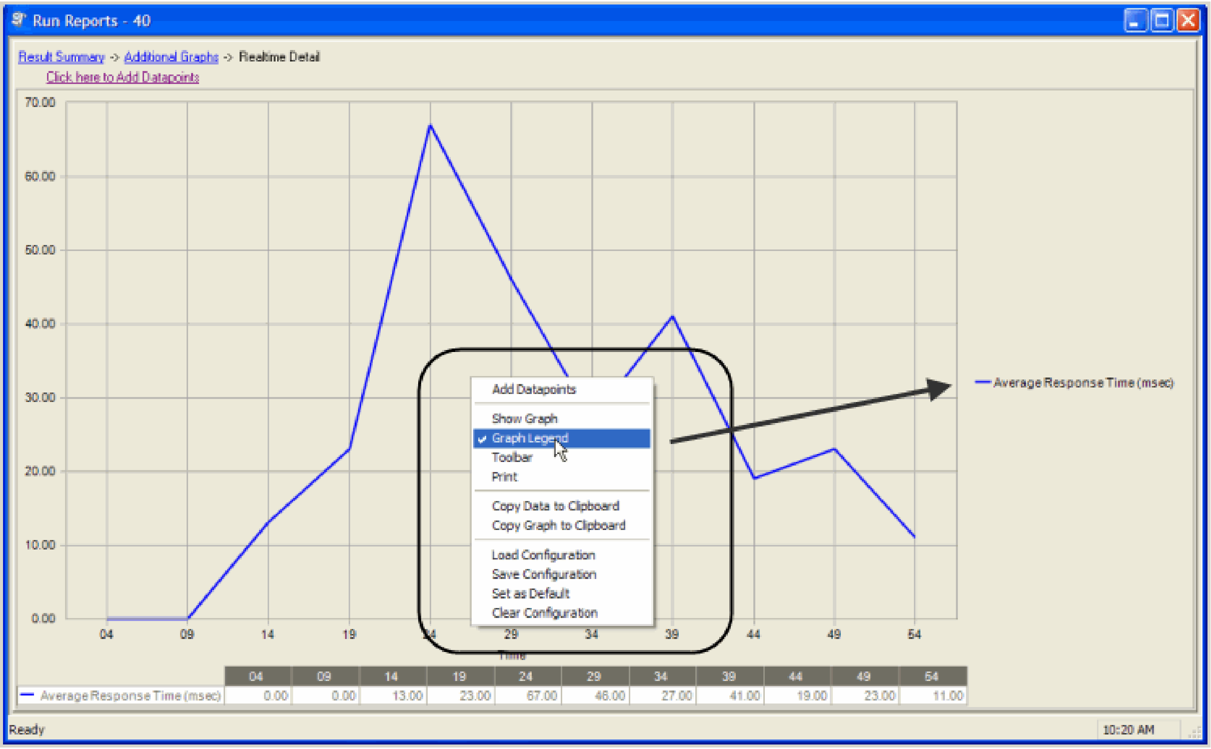
Toggling to Graph displays the data in graph form.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Graph Legend

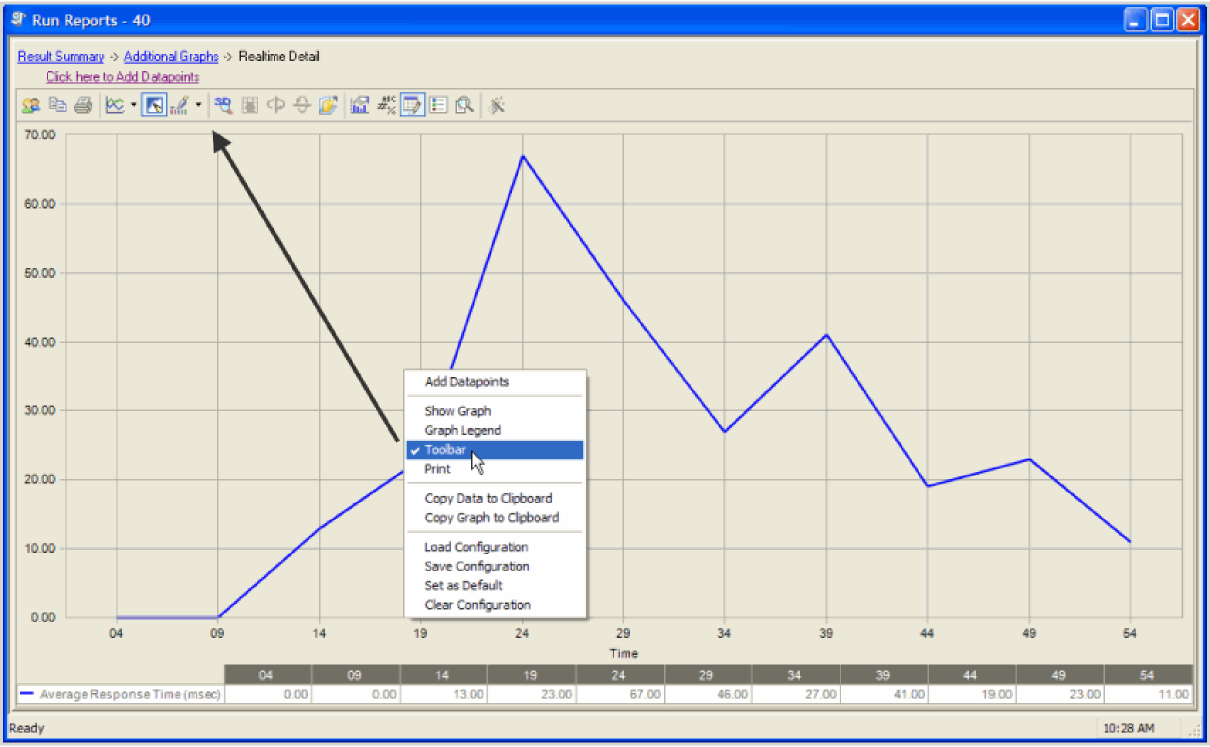
Toggling to Graph Legend displays a legend on the side of the graph.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Toolbar

Toggling to Toolbar displays the graph toolbar.



	04	09	14	19	24	29	34	39	44	49	54
Average Response Time (msec)	0.00	0.00	13.00	23.00	67.00	46.00	27.00	41.00	19.00	23.00	11.00

Print

Choosing Print displays the Print Dialog.

Copy Data to Clipboard

Copies the data to the clipboard.

Copy Graph to Clipboard

Copies the graph to the clipboard.

Load Configuration

Benchmark factories graphs allows you to save graph configurations.

Save Configuration

Saves a graph configuration.

Set as Default

Sets a configured graph as default.

Clear Configuration

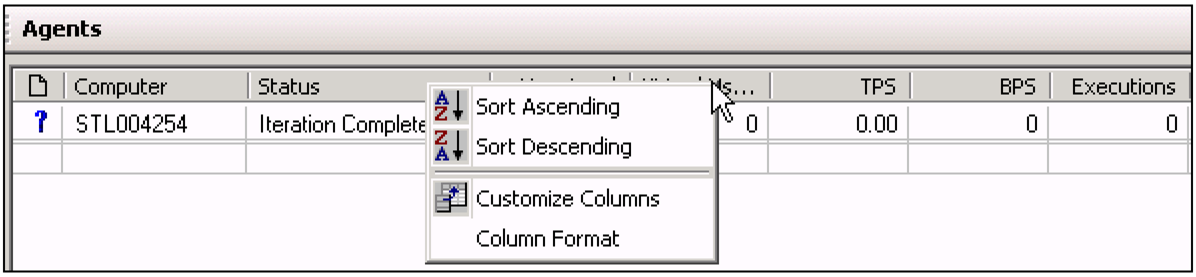
Clears a graph configuration.

Customize List Controls

Benchmark Factory allows you to customize list controls in columns to select the desired viewing format.

To Customize Lists Controls

1. Right-click the desired column. A menu displays.



	Computer	Status	s...	TPS	BPS	Executions
?	STL004254	Iteration Complete	0	0.00	0	0

2. The following options are provided:
   - **Sort Ascending**: Sorts columns in ascending order.
   - **Sort Descending**: Sorts columns in descending order.
   - **Customize Columns**: Allows you to customize column headings.
   - **Column Format**: Allows you to customize column format.

Create a SQL 7 Trace Table Using the SQL Server Profiler

To create a SQL 7 trace table

1. Initiate the SQL Profiler.
2. Click the **New Trace** icon. The Connect to SQL Server window displays.
3. Login to the SQL server. The Trace Properties window displays.
4. Type a name of a trace file into Trace Name.
5. Check the Capture To Table box. The Capture TO Table window displays.
6. Select a database name from the Database drop-down list.
7. Select dbo from the Owner drop-down list.
8. Click **OK**.
9. Click on the **Events** tab.
10. Select **TSQL** from the Available events classes list.
11. Click the **Add** button.
12. Click the **Data Columns** tab.

13. Add the following columns (minimal):
    14. EventClass
    15. SQLUserName
    16. ConnectionID
    17. NTUserName
    18. Duration
    19. StartTime
    20. Text
    21. DatabaseID
14. Click **OK**.
15. Run SQL statements.
16. When finished running SQL statements, click the **Stop this trace** icon. The table creates.

Create a SQL 2000 Trace Table Using the SQL Server Profiler

To create a SQL 2000 trace table

1. Initiate the SQL Profiler.
2. Click the **New Trace** icon. The Connect to SQL Server window displays.
3. Select the appropriate SQL Server from the drop-down list.
4. Check the **Start SQL Server if it is stopped** check box.
5. Select the appropriate Connection using radio button. Enter the appropriate login name and password.
6. Click **OK.** The Trace Properties window displays.
7. Click the **General** tab.
8. Enter a name for the trace in the Trace name field.
9. Check the **Save to table** check box. The Destination Table window displays.
10. Select the appropriate database from the Database drop-down list.
11. Select the appropriate table from the Table drop-down list.
12. Click **OK**. The Trace Properties window displays.
13. Click the **Events** tab.
14. Select TSQL from the Available events classes list.
15. Click **Add**.
16. Click the Data Columns tab.
17. Add the following columns (minimum):
    - EventClass
    - Login Name
    - ClientProcessID

- NTUserName
- Duration
- StartTime
- TextData

18. Click **Run**.
19. Run SQL statements.
20. When finished running SQL statements, click the **Stop selected trace** icon. The table is created.

Create a SQL 2005 Trace Table Using the SQL Server Profiler

To create a SQL 2005 trace table

1. Initiate the SQL Profiler.
2. Click the **New Trace** icon. The Connect to SQL Server window displays.
3. Select the appropriate server name from the drop-down list.
4. Select the appropriate authentication from the drop-down. Enter the login name and password if required.
5. Click **Connect.** The Trace Properties window displays.
6. Enter a name for the trace in the Trace name field.
7. Check the **Save to table** check box. The Connect to Server dialog displays.
8. Select the appropriate server name from the drop-down list.
9. Select the appropriate authentication from the drop-down. Enter the login name and password if required.
10. Click **Connect**. The Destination Table window displays.
11. Select the appropriate database from the Database drop-down list.
12. Select the appropriate owner from Owner drop-down.
13. Select the appropriate table from the Table drop-down list.
14. Click **OK**. The Trace Properties window displays.
15. Click the Events Selection tab.
16. Verify that all check boxes are checked in the TSQL-SQL:BatchStarting Events, and TSQL-SQL:BatchComplete Events.
17. Click **Run**.
18. Run SQL statements.
19. When finished running SQL statements, click the **Stop selected trace** icon. The table is created.

Creating a SQL 2008/2008 R2 Trace Table Using the SQL Server Profiler

To create a SQL 2008/2008 R2 trace table

1. Initiate the SQL Profiler.
2. Click the **New Trace** icon. The Connect to SQL Server window displays.
3. Select the appropriate server name from the drop-down list.
4. Select the appropriate authentication from the drop-down. Enter the login name and password if required.
5. Click **Connect.** The Trace Properties window displays.
6. Enter a name for the trace in the Trace name field.
7. Check the **Save to table** check box. The Connect to Server dialog displays.
8. Select the appropriate server name from the drop-down list.
9. Select the appropriate authentication from the drop-down. Enter the login name and password if required.
10. Click **Connect**. The Destination Table window displays.
11. Select the appropriate database from the Database drop-down list.
12. Select the appropriate owner from Owner drop-down.
13. Select the appropriate table from the Table drop-down list.
14. Click **OK**. The Trace Properties window displays.
15. Click the Events Selection tab.
16. Verify that all check boxes are checked in the TSQL-SQL:BatchStarting Events, and TSQL-SQL:BatchComplete Events.
17. Click **Run**.
18. Run SQL statements.
19. When finished running SQL statements, click the **Stop selected trace** icon. The table is created.

Oracle Instant Client Installation

To run Benchmark Factory, you must have the client dll files installed.

To install the Oracle 10g Instant Client dlls

1. Place Oracle 10g instant Client on your system following the directions provided by Oracle. For these instructions, please visit the Oracle 10g Web site.
2. Copy tnsnames.ora from a configured system to your system.
3. Edit your systems Environment Variables using one of the following methods:
    4. Add to the path variable the directory the Instant Client files were copied to.

5. Add a new System variable Named TNS_Admin, and place the directory location of the tnsnames.ora file in the value field.

Note: Environment Variables can be reached by right clicking on the My Computer Icon, selecting properties, select Advanced tab, and select the Environmental Variables button.

Migrating Repository Data Using the DOS Command Line

The Repository is a database where all test results are stored. Benchmark Factory inserts test results into the repository and provides an easy way to access the data. By default, the Repository is a MySQL or SQLite database that resides on the same machine as Benchmark Factory. The Repository can reside on another database server if required. To change the database, open the Repository Manager or use the Repository Settings page of the Setting dialog.

Benchmark Factory allows you to migrate data from the DOS command line.

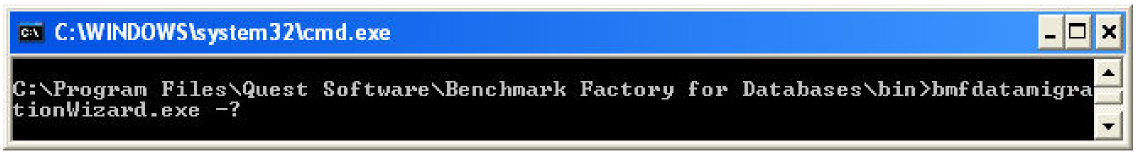
To migrate data using the DOS command line

1. Click **Start | All Programs | Accessories | Command Prompt.**
2. Navigate to the Benchmark Factory bin directory. For example:

   CD C:\Program Files\Quest Software\Benchmark Factory <*version number*>\bin.

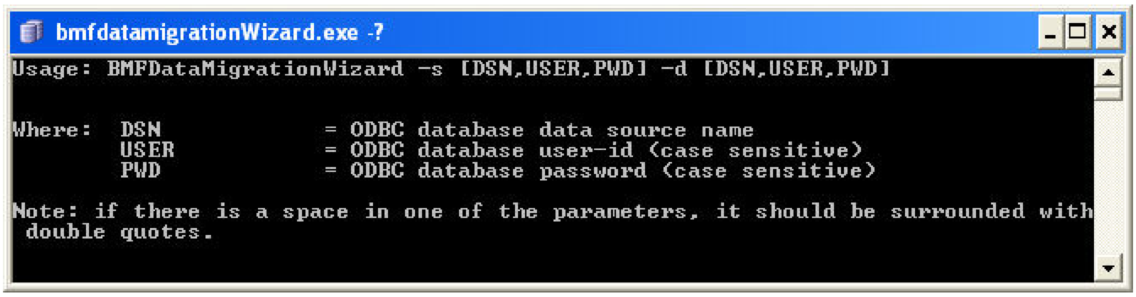
3. To display the syntax help, type:

   ```
   bmfdatamigrationWizard.exe -?
   ```

   ```
   C:\Program Files\Quest Software\Benchmark Factory for Databases\bin>bmfdatamigra
   tionWizard.exe -?
   ```

4. Press **Enter**. The usage parameters display.

   ```
   Usage: BMFDataMigrationWizard -s [DSN,USER,PWD] -d [DSN,USER,PWD]

   Where:  DSN            = ODBC database data source name
           USER           = ODBC database user-id (case sensitive)
           PWD            = ODBC database password (case sensitive)

   Note: if there is a space in one of the parameters, it should be surrounded with
    double quotes.
   ```

5. To perform the migration, use the following syntax:

   ```
   BMFDataMigrationWizard -s [DSN,USER,PWD] -d [DSN,USER,PWD]
   ```

   Example:

   ```
   BMFDataMigrationWizard -s [MyDatabase,root,yourpassword] -d[LocalServer,
   sa,sa]
   ```

Store Procedure Examples

The following provides syntax examples of statements that call stored procedure. Click here for procedures on creating a load scenario by importing stored procedures.

<table>
<tr><th>Database</th><th>Examples</th></tr>
<tr><td><b>Oracle</b></td><td><b>Example with parameters</b>:

BEGIN
YOUR_PROC(:VAR1,:VAR2);
END;

<i>Bind Parameter Settings</i>

<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>:VAR1</td><td>INTEGER</td><td>5</td></tr>
<tr><td>:VAR2</td><td>INTEGER</td><td>8</td></tr>
</table>

<b>Example without parameters:</b>

BEGIN
YOUR_PROC2 ( );
END;</td></tr>
<tr><td><b>Microsoft SQL Server and SAP</b></td><td><b>Example with parameters</b>:

exec YOUR_PROC(@VAR1,@VAR2)

<i>Bind Parameter Settings</i>

<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>@VAR1</td><td>INTEGER</td><td>55</td></tr>
<tr><td>@VAR2</td><td>INTEGER</td><td>77</td></tr>
</table>

<b>Example without parameters:</b>

exec YOUR_PROC2</td></tr>
<tr><td><b>IBM DB2, ODBC, and MySQL</b></td><td><b>Example with parameters</b>:

call procedure-name (?,?,?,?);

<i>Bind Parameter Settings</i>

<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>1</td><td>STRING</td><td>A</td></tr>
<tr><td>2</td><td>INTEGER</td><td>5</td></tr>
</table>
</td></tr>
</table>

3	INTEGER	7
4	INTEGER	0

Example without parameters:

call procedure-name ();

Note: When using an ODBC connection, the syntax for the statement will follow the syntax for the database, with the exception of the parameters. The parameters within the statement would be "?" and the bind parameters names should be 1,2,3,4...

Oracle Trace File Activation

Benchmark Factory allows you to import an Oracle trace files for use during database testing. The following provides Oracle commands that can be used to enable Oracle trace file activity.

Note: Please consult Oracle documentation for current trace file commands.

To turn on and off the trace use the ALTER SESSION command:

1. Set the TIMED_STATISTICS and MAX_DUMP_ FILE_SIZE parameters used by the sessions:

   ```
   alter session

   set timed_statistics=true

   alter session

   set max_dump_file_size=unlimited
   ```

2. Activate the trace:

   ```
    alter session set events
   '10046 trace name context forever, level 4'
   ```

3. `Stop the trace:`

   ```
    alter session set events

   '10046 trace name context off'
   ```

To activate a trace you don't have read/write access to:

1. Acquire the V$SESSION.SID and V$SESSION.SERIAL# values of the session you wish to trace.
2. Set the TIMED_STATISTICS and MAX_DUMP_FILE_SIZE parameters for the session utilizing the following procedure calls:

   ```
   dbms_system.set_bool_param_in_session(

      sid => 55,

      serial# => 1222,

      parnam => 'timed_statistics',

      bval => true)

   dbms_system.set_int_param_in_session(

      sid => 55,
   ```

```
   serial# => 1222,

   parnam => 'max_dump_file_size',

   intval => 2147483647)
```

Note: If using a release before Oracle 8, release 8.1.6, these parameters can be changed with ALTER SYSTEM commands.

3. Activate the trace.

```
dbms_support.start_trace_in_session(
   sid => 55,
   serial => 1222,
   waits => true,
   binds => true)
```

4. Stop the trace.

```
dbms_support.stop_trace_in_session(
  sid => 55,
  serial => 1222)
```

System Requirements/Upgrade Requirements/Supported Databases

System Requirements

Before installing Benchmark Factory, ensure your system meets the following minimum hardware and software requirements.

Requirement	Details
Repository	Benchmark Factory cannot run without a repository, and during installation a SQLite file-based repository is installed. After installation, you can select a repository of your choice. Benchmark Factory supports the following databases: • Oracle® • SQL Server® • MySQL • SQLite • SAP SQL Anywhere® (Adaptive Server® Anywhere)
Memory	512 MB of RAM recommended **Note:** The memory required may vary based on the following: • Applications that are running on your system. • Size and complexity of the database. • Amount of database activity. • Number of concurrent users accessing the database.
Platform	1.0 GHz x86 or x64 compatible CPU
Browser Requirements	Benchmark Factory requires Internet Explorer® 9 or later.

Requirement	Details
Hard Disk Space	120 MB of disk space for installation Note: The disk space required may vary based on options selected for installation.
Operating System	Microsoft® Windows Server® 2008 Service Pack 2 Microsoft Windows Server 2008 R2 Service Pack 1 Microsoft Vista® Service Pack 2 Microsoft Windows® 7 Service Pack 1 Microsoft Windows 8 Microsoft Windows 8.1 (Update) Microsoft Windows Server 2012 Microsoft Windows Server 2016 Microsoft Windows 10
.NET Framework	.NET Framework 2.0 is required to use Oracle workloads captured using Benchmark Factory 6.9.3 or earlier.
Additional Software	The Universal C Runtime component for Windows is required. See Universal C Runtime Update or Visual C++ Redistributable for Visual Studio 2015 to download this software. **Note:** If you encounter an error when installing this software, install the missing prerequisite software. For Windows 8.1 or Windows Server 2012 R2, install the April 2014 update: https://support.microsoft.com/en-us/kb/2919355. For other operating systems, see the Universal C Runtime Update Prerequisites section.
SQL Server Native Provider	For a list of operating systems that support the SQL Server native provider distributed with Benchmark Factory (native client 11.0), see https://msdn.microsoft.com/en-us/library/cc280356(v=sql.120).aspx.
Database Server	Oracle®: 9i, 9.2, 10g, 11g, and 12c (requires Oracle 9 client libraries or higher.) Benchmark Factory has been tested on Oracle Exadata 2.0 running Oracle database 11g R2. ℹ **Note**: Oracle Capture/Replay of workloads must use Oracle 9i or higher. SQL Server®: 2000 (ODBC only), 2005, 2008, 2008 R2, 2012, 2014, 2016, and 2017 • Benchmark Factory has been tested against SQL Server 2017 running on Windows or Linux. SQL Azure™: Benchmark Factory provides limited support of SQL Azure. SAP® ASE: 12.5, 15, 15.5, 15.7, 16 IBM® DB2® for LUW: 9.x through 10.x, 11.1 IBM® DB2® for z/OS: 9.x through 10.x, 11.1 MySQL: 4.5 and 5.x ℹ **Note**: The 32-bit version of Benchmark Factory requires the 32-bit versions of the client libraries (including ODBC drivers). The 64-bit version requires 64-bit client libraries.
ODBC Database Server	Benchmark Factory supports almost all databases that you can connect to using an ODBC 3.0 or later driver.
Cloud Database Service	Oracle Database as a Service

Upgrade Requirements

- Client libraries for database types used during the workload testing process must be installed on all testing machines (Benchmark Factory and Agents).
- There is no upgrade path for the Benchmark Factory Repository version 3.3 or earlier.
- If you create a new Benchmark Factory 5.5 or later repository, earlier versions of Benchmark Factory will not work against this repository.

Shortcut Keys

The following provides a list of shortcut keys used in Benchmark Factory.

Key	Action
ALT+1	Displays the Output View window
ALT+2	Displays the Agents View window
ALT+M	Creates an email message with the current script attached
ALT+R	Runs a job
ALT+S	Stops a job
CTRL+B	Displays the Benchmark Objects Wizard
CTRL+C	Copies the selected item in the Jobs View
CRTL+D	Saves a job back to the Jobs View
CTRL+E	Executes a job
CTRL+N	Displays the New Job Wizard (Jobs View only)
CTRL+O	Displays the Select Script File to Open dialog (Jobs View only)
CTRL+P	Displays the Edit Connections dialog
CTRL+R	Displays the Enter a New Name dialog (Jobs View only)
CTRL+S	Saves the current script (Jobs View only)
CTRL+V	Paste an item from the clipboard (Jobs View only)
CTRL+W	Displays the New Connection dialog
CTRL+X	Cuts a selected item (Jobs View only)
CTRL+ ALT+E	Readies all jobs
CTRL+H	Holds a job
CTRL+ALT+H	Holds all jobs
Delete	Deletes the selected item (Jobs View only)
END	Selects the end of the item with focus
HOME	Selects the beginning of the item with focus
F1	Displays online help
F8	Displays the Setting dialog
F12	Displays the Save As dialog

Creating an ODBC Trace File

To create an ODBC trace file

1. Click the **Start** button from the Windows taskbar.
2. Select **Settings | Control Panel**. The Control Panel window displays.
3. Double-click on **Administrative Tools**. The Administrative Tools window displays.
4. Launch **Data Source (ODBC)**. The ODBC Data Source Administrator window displays.
5. Click the **Tracing** tab.
6. Click the **Start Tracing Now** button. The Start Tracing Now button changes to Stop Tracing Now button.
7. Run the ODBC transactions to be traced.
8. When the transactions complete, click on the Stop Tracing Now button.
9. Click **OK**.

Configure Firewall for Remote Agent Install/Start-Up

In order to install remote agents through the Benchmark Factory console, you need to turn off the firewall or configure the firewall to allow traffic between the console machine and the agent machine. To allow traffic thru the firewall, enable Windows Management Instrumentation (WMI) and open TCP port 135 on the agent machine.

In addition, after the remote agent is installed, the Benchmark Factory console and the remote agent also communicate through port 4568 (by default). To enable this communication, configure an inbound rule on the console machine and an outbound rule on the agent machine.

Note: The console and remote agent communicate through port 4568 by default. You can change this port number in **Edit | Settings | General**. Check this setting to ensure you use the specified port number when creating inbound/outbound rules.

Enable WMI on Agent Machine

Enable Windows Management Instrumentation (WMI) to allow communication between the Benchmark Factory console and the remote agent. The following instructions are applicable to Windows 7. For more information, see http://msdn.microsoft.com/en-us/library/aa822854(v=vs.85).aspx.

To enable WMI (Windows 7)

1. Select **Control Panel | System and Security | Windows Firewall**.
2. Click **Allow a program or feature through Windows Firewall**.
3. Select the checkbox to the left of **Windows Management Instrumentation (WMI)**.
4. Click **OK**.

Set Inbound and Outbound Rules

The Benchmark Factory console uses TCP port 135 to start agents remotely, and port 4568 is used by the console and the agent machine for communication. Create inbound and outbound rules to allow port 135 and port 4568 traffic through the Windows Firewall.

Create Inbound Rule on console machine

1. Select **Control Panel | System and Security | Windows Firewall**.
2. Click **Advanced Settings**. The Windows Firewall and Advanced Security dialog opens.
3. Click **Inbound Rules** (left panel).
4. Click **New Rule** under Actions (right panel).
5. Select **Port** in the New Inbound Rule Wizard. Click **Next**.
6. Select **TCP** and **Specific local ports**.
7. In the **Specific local ports** text box, enter "*4568*" to add these ports. Click **Next**.
8. Select **Allow this connection**. Click **Next**.
9. Select **Domain**, **Private**, and **Public**. Click **Next**.
10. Give your new rule a name and a description. Then click **Finish** to add the rule.
11. Your new rule appears in the Inbound Rules list.

Create Inbound Rule on agent machine

- On the agent machine, repeat the steps listed above for creating an inbound rule. However, in the **Specific local ports** text box, enter "*135*".

Create Outbound Rule on agent machine (if outbound connections are blocked)

1. Select **Control Panel | System and Security | Windows Firewall**.
2. Click **Advanced Settings**. The Windows Firewall and Advanced Security dialog opens.
    - If outbound connections are blocked, then continue to create a new outbound rule.
    - If outbound connections are allowed, then no action is required.

      Note: In Windows Firewall, outbound connections are set to “Allow” by default.
3. If outbound connections are blocked, click **Outbound Rules** (left panel).
4. Click **New Rule** under Actions (right panel).
5. Select **Port** in the New Outbound Rule Wizard. Click **Next**.
6. Select **TCP** and **Specific local ports**.
7. In the **Specific remote ports** text box, enter "*4568*" to add this ports. Click **Next**.
8. Select **Allow this connection**. Click **Next**.
9. Select **Domain**, **Private**, and **Public**. Click **Next**.
10. Give your new rule a name and a description. Then click **Finish** to add the rule.
11. Your new rule appears in the Outbound Rules list.

Troubleshooting

After enabling WMI and configuring inbound/outbound rules, if you encounter an error while attempting to install a remote agent because you are denied access, try the following.

Enable access using the command line (Windows Vista)

- Open a Command window and enter the following:

  ```
  netsh firewall set service RemoteAdmin enable
  ```

Add a Registry key (Windows Vista, Windows 7, Windows 8)

1. Open the Registry Editor (regedit.exe).

   ! **Caution: Editing your Registry incorrectly can cause serious, system-wide problems. It is advised that you back up your Registry before modifying it.**

2. Navigate to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System.
3. Add a new DWORD (32-bit) Value.
4. Rename the key to "LocalAccountTokenFilterPolicy".
5. Give it a value of "1".
6. Close the Registry Editor.

About Us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contact Quest

For sales or other inquiries, visit www.quest.com/contact.

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

Benchmark Factory Community

Get the latest product information, find helpful resources, and join a discussion with the Benchmark Factory team and other community members. Join the Benchmark Factory community at http://www.toadworld.com/products/benchmark-factory/default.aspx

Index