

Verax Database Load Simulator - User Guide

October 2011

Version 1.0.2



Contact Information:

E-mail: sales@veraxsystems.com

Internet: <http://www.veraxsystems.com>

Technical support:

E-mail: support@veraxsystems.com

COPYRIGHT AND DISCLAIMER

Copyright © Verax Systems. All rights reserved.

Verax Systems have taken care in the preparation of this publication, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

All brand names or product names mentioned in this publication are either trademarks or registered trademarks of their respective owners.

TABLE OF CONTENTS

How to use this guide?	4
Purpose and scope.....	4
Notation used.....	4
Intended audience and guide overview.....	4
1 Introduction	6
1.1 How does it work?	6
1.2 Supported databases	7
1.3 Supported operating systems.....	7
2 Installation	8
3 Running	9
3.1 Command line parameters	10

How to use this guide?

Purpose and scope

This user guide contains description of the installation, configuration and management procedure for Verax Database Load Simulator, a test tool for generating database load by executing random SQL queries.

Notation used

Source code, commands, user-entered data, on-screen messages and user interface elements (menus, choice lists, etc.) are shown using `Courier font`. In order to improve readability, indentation has been used, for instance:

```
int main() {  
    int i = 0;  
}
```

! This notation (**Information**) is used to indicate important information.

⚠ This notation (**Warning**) is used to flag actions that can lead to data loss, system malfunction, etc.

ℹ This notation (**Hint**) is used to indicate additional information.

The following logotypes are used to flag information relevant to a particular operating system:



Linux



Oracle
Solaris



Microsoft
Windows



IBM AIX

Intended audience and guide overview

This user guide is intended for database administrators and/or software developers who wish to install and run the Verax Database Load Simulator.

The guide consists of the following sections:

- **Section 1, Introduction** describes what Verax Database Load Simulator is, how it works and its compatibility with various databases.
- **Section 2, Installation** describes how to install the simulator from prerequisites to the first run.
- **Section 3, Running** describes how to start and use the simulator.

1 Introduction

Verax Database Load Simulator is a command line stress test tool written in the Java programming language that can be used to generate high database loads by executing SQL queries for INSERT, UPDATE and DELETE operations. The simulator can be used to test database performance or (more often), application performance when its database is under high load conditions. The simulator is used internally at Verax Systems for stress testing of our products.

1.1 How does it work?

The simulator connects to the database using a specified JDBC connection string and creates three tables – test1, test2 and rel. The first two tables the following elements: `id` (a unique key for each row), `name` (a random string of letters) and `float_value` (a random floating point value). The third table joins elements from the others and consists of two columns – `id_test1` and `id_test2`.

Once the tables have been created, the simulator performs a random number (the maximum being specified as the `<limit>` command line parameter) of randomly selected operations (INSERT, UPDATE or DELETE).

Finally, the simulator sleeps for the time specified by the `<pause>` command line parameter and restarts the INSERT, UPDATE, DELETE cycle. The simulator runs in an infinite loop until stopped from the command line using Ctrl-C.

1.2 Supported databases

The Verax Database Load Simulator has been successfully tested with the following databases:

- MySQL 5.1
- Oracle 10g Express Edition
- PostgreSQL 9.0
- Microsoft SQL Server 2008

Since the simulator uses JDBC and general SQL queries only, it should be compatible with most database systems.

1.3 Supported operating systems

The simulator will run on any operating system running Java 1.6 or higher including Windows, Linux, Solaris, AIX and others.

2 Installation

Prerequisites

The simulator requires JRE version 1.6 be installed on the system. A JDBC driver for the database used has to be installed (this usually involves copying an appropriate driver to the java CLASSPATH, please refer to database documentation for specific details).

Installation procedure

In order to install Verax Database Load Simulator:

1. Download `vxdatabaseloadsimulator-1.0.2.zip` file.
2. Unzip the file.
3. Move content of this archive (`DatabaseLoadSimulator.jar` and `drivers` directory) to program files directory e.g.:

WIN

`C:/Program Files/DatabaseLoadSimulator`

LINUX

`/usr/local/DatabaseLoadSimulator`

4. Copy required JDBC drivers to the `drivers` directory.
5. Make sure that environment variables for Java are set. For more details see:
<http://www.java.com/en/download/help/path.xml>

3 Running

In order to start Verax Database Load Simulator:

1. Open the command line shell in the installation directory:

WIN

Open the Command Prompt

Change directory to: C:\Program Files\DatabaseLoadSimulator

LINUX

Open terminal window

Go to /usr/local/DatabaseLoadSimulator directory.

2. Run the simulator, using the following command (see section 3.1 for details on command line parameters):

```
java -jar DatabaseLoadSimulator.jar -Djava.library.path=drivers  
<connection_string> <threads> <pause> <limit>
```

●* All the parameters must be specified in order exactly as presented above.

3. If all parameters are provided correctly the following line should appear:

```
Verax Systems Database Load Simulator v1.0.2. Copyright (c)  
Verax Systems. All rights reserved.
```

```
Simulator started successfully.
```

4. Press CTRL-C and confirm by pressing 'Y' to stop the simulator.

3.1 Command line parameters

The meanings of command line parameters are as follows:

`-Djava.library.path` - path to the JDBC drivers directory,

`<connection_string>` - the JDBC connection string, identifying database connection. The connection string format varies depending on a particular database type:

Database	<connection string> syntax
MySQL	<pre>jdbc:mysql://<HOST>:<PORT>/<DB></pre> <p>For instance:</p> <pre>java -jar DatabaseTrafficSimulator.jar "jdbc:mysql://192.168.100.70:3306/tempdb?user=admin &password=mypass" 3 500 100</pre>
Oracle	<pre>jdbc:oracle:thin:@<HOST>:<PORT>:<SID></pre> <p>For instance:</p> <pre>java -jar DatabaseTrafficSimulator.jar "jdbc:oracle:thin:admin/mypass@192.168.104.33:1521:XE" 6 800 200</pre>
PostgreSQL	<pre>jdbc:postgresql://<HOST>:<PORT>/<DB></pre> <p>For instance:</p> <pre>java -jar DatabaseTrafficSimulator.jar "jdbc:postgresql://192.168.39.11:5432/tempdb?user= admin&password=mypass" 2 1000 2000</pre>
Microsoft SQL Server	<pre>jdbc:sqlserver://<HOST>:<PORT>/<DB></pre> <p>For instance:</p> <pre>java -jar DatabaseTrafficSimulator.jar "jdbc:sqlserver://192.168.75.20;databaseName=tempdb; user=admin;password=pass" 8 100 5000</pre>

<threads> - Number of threads spawned by the simulator. Each thread has its own database connection. The limit of operations specified by the <limit> parameter has a per thread meaning – larger number of threads means higher database load.

<pause> - Pause time between loops (in milliseconds).

<limit> - Limit of operations made in a single loop.