Sterling Commerce

Platform SQL Proxy Tool

Platform IFC 1.0

Sterling Commerce
An IBM Company

SQLProxy Tool

What is the SQLProxy Tool?

SQLProxy captures SQL statements going through a JDBC driver to the database and generates JDBC traces. SQLProxy captures all SQL statements, regardless of the type of database being used. Another tool, the SQL Proxy Analyzer, produces a report based on the SQLProxy trace records. Using the SQLProxy tools can help you to analyze database response times for queries, which can help your capacity planning and maximize efficient usage of your database servers. See *Analyzing the Results* on page 4 for more information about the Analyzer.

Setting Up the SQLProxy Tool

1. In the \<INSTALL_DIR>\properties\ folder, locate the yfs.properties file, and edit it to set the following properties before starting WLS:

Property	**Description**
yfs.enable.proxy.sql.logging=Y	Determines if logging is needed
yfs.enable.source.logging=Y	Specifies if line of code needed in the output. Useful while providing trace for developer but consumes slightly more time.
yfs.proxy.log.dir=/home/test03/sqllogs	Directory where the trace files will be generated.

2. In the System Management Console turn on TIMER trace for the desired component (for example, the API or agent). See “Tracing a Component” in the *System Management Guide* for information about the types of components that can be traced and instructions for starting a trace.

Setting Up the SQLProxy Logging for NoAppServer

In the \<INSTALL_DIR>\properties\ folder, locate the jdbc.properties file, edit it and do the following before starting NoAppServer:

1. Append *jdbc:sci:* to the start of the JDBC URLs. For example, the JBDC URL *jdbc:mysql://<URL>* will change to *jdbc:sci:jdbc:mysql://<URL>*.

   For the format of the JDBC URL, refer to your database vendor documentation or your JDBC driver documentation.

2. Change the *.driver properties to com.yantra.jdbc.driver.SCIProxyDriver. For example, *mysqlPool.driver=com.mysql.jdbc.Driver* property will change to *mysqlPool.driver=com.yantra.jdbc.driver.SCIProxyDriver.*

3. Set the following properties:

Property	Description
proxyLoggingEnabled=Y	Determines if logging is needed
proxySourceLogging=Y	Determines if logging is needed
proxyDriver=<Set to your JDBC Driver class>	Specify the name of the JDBC driver class.
proxyLogDir=<INSTALL_DIR>\logs	Directory where the trace files will be generated.

Note: When you set up SQLProxy Logging using the SCIProxyDriver driver, make sure that you do not enable the SQLProxy Logging by setting up the properties (such as yfs.enable.proxy.sql.logging, yfs.enable.source.logging, and yfs.proxy.log.dir) in the yfs.properties file.

Starting and Using the SQLProxy Tool

To begin tracing, execute the API or agent. The trace files are generated in the `yfs.proxy.log.dir` directory. The following details are included in the files:

- *.log file gives component name, start and end system times, operation execution times (individual and cumulative), operation name, SQL statement, java class name
- *.tail file helps in understanding the long running queries (more used in performance testing)

Stopping the SQLProxy

Turn off the SQLProxy by deleting the directory where the trace files are generated. If needed again, recreate the directory. See “Stopping a Component Trace” in the *System Management Guide*.

Analyzing the Results

You can export the contents of a file to a .csv file and open it in XSL. There is also a support tool called the SQLProxy Analyzer which analyzes and summarizes JDBC trace records. This tool is available as part of MTCP in Eclipse.

Caution: Contact Sterling Commerce Customer Support when using this tool.

Before using the SQLProxy Analyzer, consider the following limitations:

- Prepared Statements are not currently being captured by the proxy analyzer
- This feature does not work with JDBC connection pool when integrated with Sterling applications (However, we can choose not to go for connection pooling in our environments)

About the SQLProxy Analyzer

The SQLProxy Analyzer is a tool used to analyze the data in the SQLProxy logs. The Analyzer has the following capabilities:

- Can process multiple log files in a directory or a single log file
- For each log file (which represents a database connection), the Analyzer produces a summary of the unit of work (UOW). (There might be multiple UOWs in a connection.)
- Can calculate the number of executions and total cost of the SQL
- Response Time Buckets, which are useful if you have a many SQL instances and a high average SQL time. The response time bucket will show you which bucket the SQL response time falls into. \
- Provides execution plans, which are automatically generated by the Analyzer. You can specify any Oracle database (it does not necessarily have to be the one from which the SQL was captured). This enables you to point the analyzer to a very large database that has up-to-date Oracle statistics.
- For each unit of work, the Analyzer provides:
  - A summary of all the SQL (SQLs are grouped together to get a small summary from a very large file)
  - The LOCK HOLDING time (therefore, you can see if the system grabs locks that could impact other workloads)
  - Oracle EXPLAIN plans, which assist you in assessing whether certain queries will potentially get slower as the database grows larger