PREPARING A TRACE FILE USING SQL PROFILER

The trace file is the primary source for the Qure's analysis of the database. Qure assumes that the trace file(s) that are provided reflect the full range of activities that the application and database perform.

To get the most effective recommendations from Qure's analysis, it is crucial that you adhere to the following guidelines when preparing and selecting the trace file(s) to be used by Qure.

The following topics explain how to prepare a trace file for the Qure database analysis process:

1 USING THE PROFILER TEMPLATE

It is highly recommended that you use the provided Profiler template for generating the trace files. Although Qure will accept any trace file that complies with its minimal requirements in terms of traced events and data columns, using the provided template will reduce the probability of errors, overinflated trace files with unused data, missed events, and so on.

➢ **To use the Profiler template:**

1. In Windows Explorer, navigate to the Qure installation folder and then to the **templates** sub-folder.
2. Double-click the **Qure_For_SQL_2005.tdf** or **Qure_For_SQL_2008.tdf** file according to the version of Profiler that you use. This imports the Qure template into the Profiler's available templates list.
3. Launch Profiler, and then click **New Trace** or select **File > New Trace**.
4. Connect to the production database server.
5. From the **Use the template** drop-down list, select the Qure template.

IMPORTANT: Do not select the **Save to file or Save to table options** before starting the trace as these options may significantly increase the load overhead that Profiler generates on the production server. Always save the trace to file only after you have completed and stopped the trace.

6. Before starting the trace, read the rest of the guidelines thoroughly.

➢ **To perform a server-side trace using the provided templates:**

Warning: On highly loaded systems, using Profiler may generate a severe performance overhead. If you use Profiler and notice performance degradation on the production server during the trace, it is advised to stop the trace immediately and revert to using a server-side trace. Server-side traces, although slightly more complex to use, usually generate considerably less performance overhead than Profiler for generating the trace files!

7 Import the provided template into Profiler, as described in the previous procedure.

8 Add the appropriate filters according to the guidelines described below.

In Profiler, **Select File > Export > Script trace definition > For SQL Server 2005–2008**. Save the trace script and use it to perform the server-side trace.

Note: For more information on performing a server-side trace, see SQL Server Books On Line.

2 ADDING FILTERS TO THE TRACE

Add filters to the trace template to achieve the following goals:

- Reduce trace file size.
- Reduce the negative effect that the tracing process may have on the production environment.
- Reduce the time required for Qure analysis process.
- Increase the efficiency of the Qure trace search algorithms.

The most important filter is on the analyzed Database ID. If the trace file contains events that are in the context of any database other than the one selected for analysis, these are ignored. The Database ID filter should be manually added to the trace definition.

Additional filters can be added at your discretion.

➢ **To add a filter on the Database ID to the trace:**

9 Find the Database ID by using the following query:

```
SELECT DB_ID(N'<Database_Name>')
```

10 In Profiler, in the Trace Properties window, click the **Events** tab.

11 Click **Column Filters**. The Edit Filter window appears.

12 In the left pane, select the **DatabaseID** column.

13 In the right pane, expand **Equals** by clicking on the [+] icon and enter the Database ID.

14 Click **OK** to close the Edit Filter window.

Qure does not analyze or provide recommendations for system activities (such as executing SQL Server system procedures), replication activities, maintenance activities (such as backups and statistics updates), and security activities (such as logins, permission changes), and so on.

You can safely filter these activities out according to your application-specific behavior. We recommend that you perform a short sample trace and manually analyze it to see which events can be safely filtered out.

For example, here are a few filters you may want to consider. You can use any of the trace data columns to apply the filters:

- Filter out SQL Agent activities using a filter on the application name, unless you use SQL Agent jobs to execute application procedures.
- Filter out ADO internal operation activities using filters on text data, such as "`EXEC sp_reset_connection`", "`IF @@trancount > 0 COMMIT TRANSACTION`", and so on.
- Filter out Management Studio activities using a filter on the application name unless you want to deliberately execute specific activities manually to be included in the trace file.

3 ADDITIONAL TRACE FILE RECOMMENDATIONS

Do not use filters based on resource consumption such as duration, reads, writes, or CPU. Filtering out any "real" application activities during the trace will have an effect on Qure's final recommendations.

For example, Qure considers the table activity when evaluating index recommendations. Filtering out "`INSERT… VALUES…`" statements, which usually consume very few resources and which cannot be explicitly optimized, may lead to Qure assessing that the table has fewer modifications than in reality, leading to less optimal index recommendations.

If you use a backup of the production database, restore it using its original name. While loading a trace file, Qure analyzes the first 2000 batches and looks for events that are executed in the context of the same database name as the one selected.

If no matching name is found, Qure assumes that it was restored using a different name and a window appears with the names of the databases found in the trace. Qure allows you to choose one of these databases and creates a constant mapping of the selected name from the trace to the database name chosen for analysis.

Qure attempts to replace the original database name found in the trace with the database name selected. However, this practice is not recommended. In some cases, Qure cannot make the correct replacement in server-side code such as stored procedures or functions that reference object names that include the database name.

It is therefore strongly recommended that you use the original name (used in production) when restoring the database to the analysis environment, eliminating the need for name mapping.

When using a backup of the production database as the target for analysis, prepare the trace files soon after the backup is made. This enables Qure to correctly replay and analyze the queries from the trace file against the data in the database during the benchmark process.

For example, if you use an old backup with a recent trace file, or vice versa, the queries may return different results than they would in production due to data mismatches; this in turn may result in a less accurate benchmark results.