

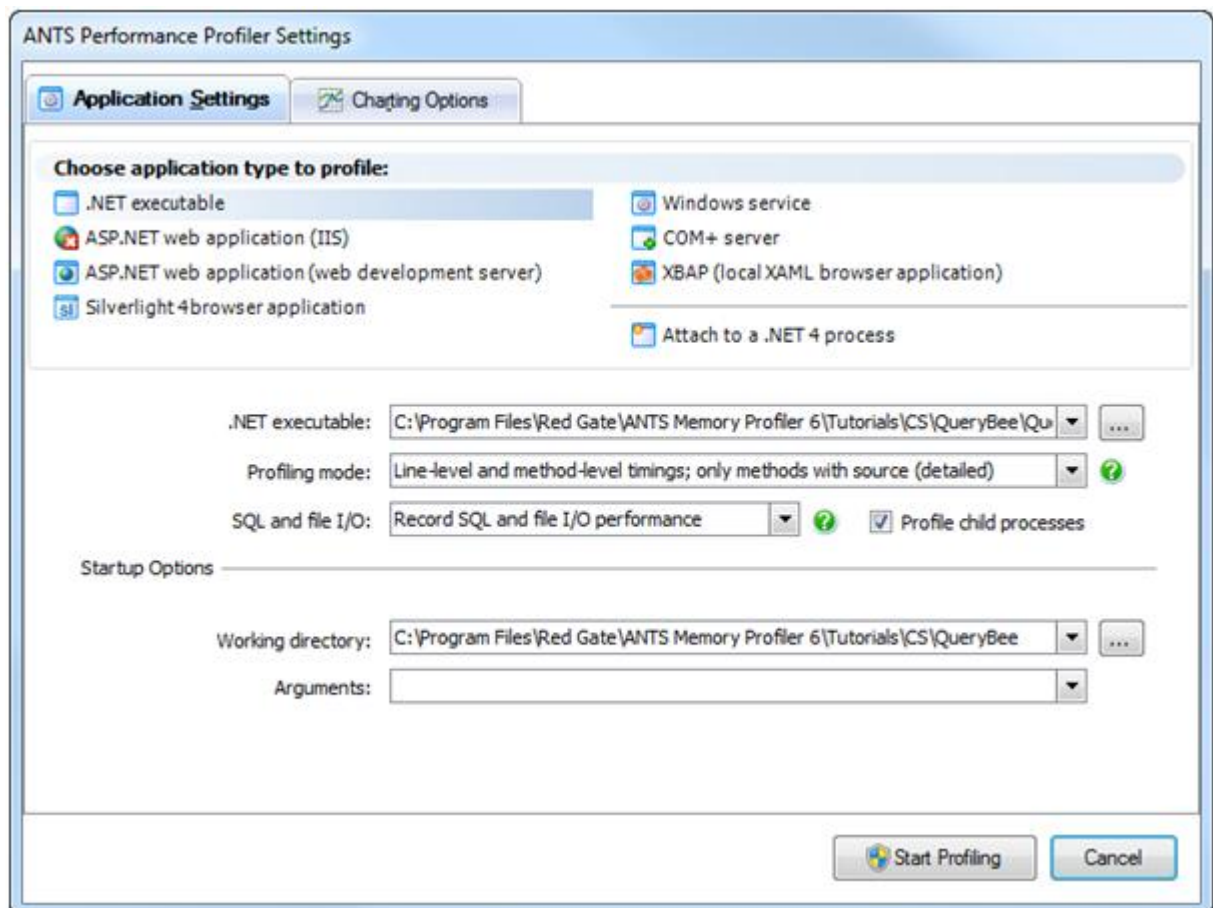
## Profiling SQL with ANTS Performance Profiler 6

ANTS Performance Profiler 6 adds SQL Profiling and File I/O to the many performance counters already available in ANTS Performance Profiler 5.

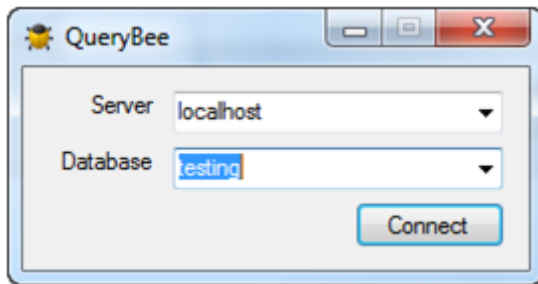
This technical paper demonstrates how you can profile the SQL queries generated by your application. A demonstration database table called 'Customers' has been populated with 100,000 records using Red Gate's SQL Data Generator (a tool designed to populate database tables with realistic data suitable for use in testing). The table has an ID column called 'ID1' but no primary key or other indices.

SQL queries are entered manually in QueryBee, a simple program for querying SQL databases, and profiled. Profiling an application which generates the queries itself is possible using exactly the same procedure.

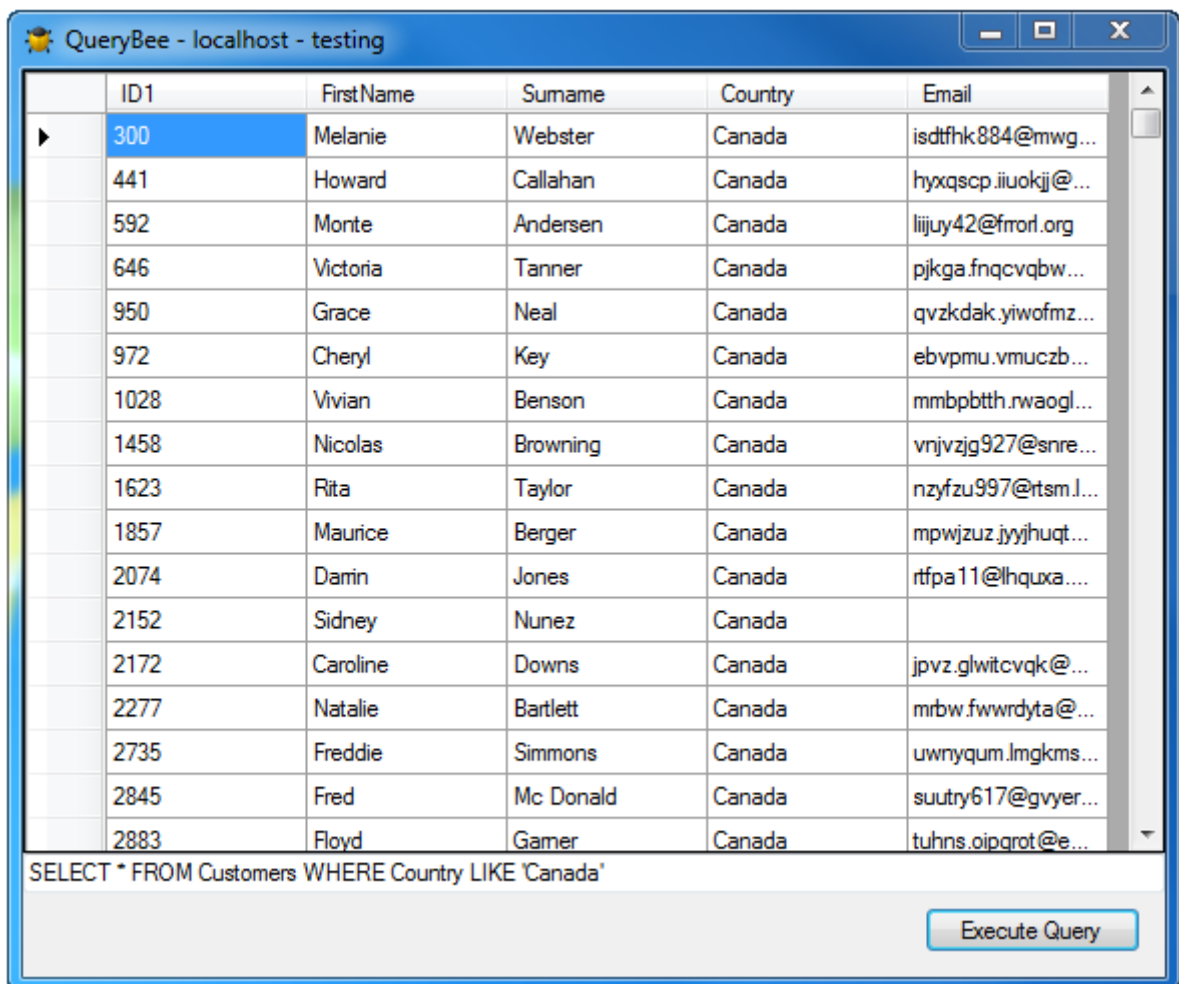
1. Set-up ANTS Performance Profiler to profile QueryBee. This is the application that will send the SQL queries to the SQL server.
2. Set the path to QueryBee in the **ANTS Performance Profiler Settings** dialog, and ensure that **Record SQL and file I/O performance** is enabled.



- Click **Start Profiling** to start QueryBee. Select the test database and click **Connect**.

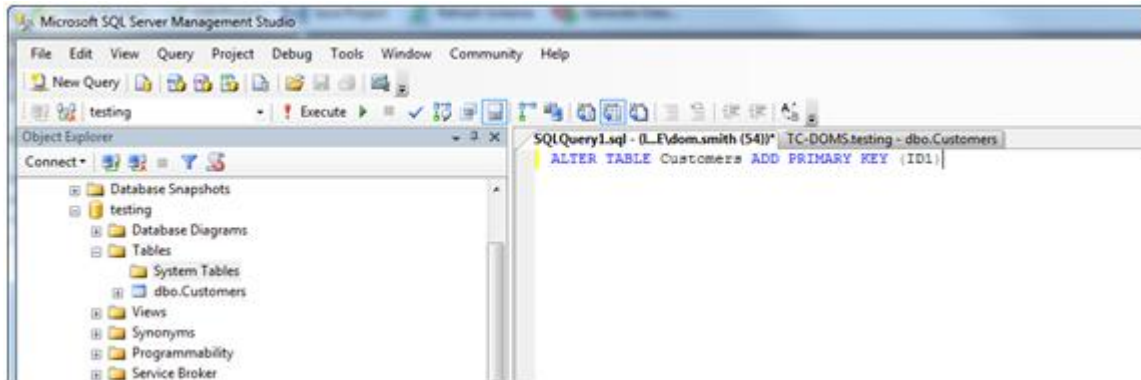


- Run the query `SELECT * FROM Customers` to ensure that all records are read into memory, and that further queries will not include file I/O overheads.
- Run the query `SELECT * FROM Customers WHERE Country LIKE 'Canada'`

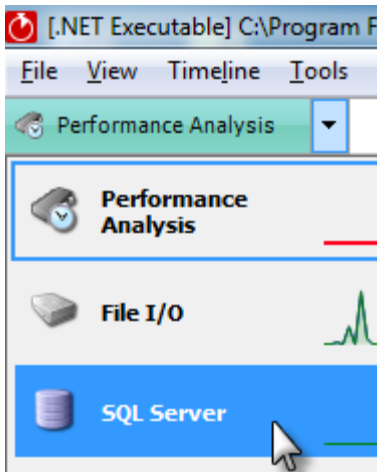


- The following query is executed, using the equals operator instead:  
`SELECT * FROM Customers WHERE Country='Canada'`
- To test updated information, run the following query:  
`UPDATE Customers SET Email='a@example.com' WHERE ID1=100`

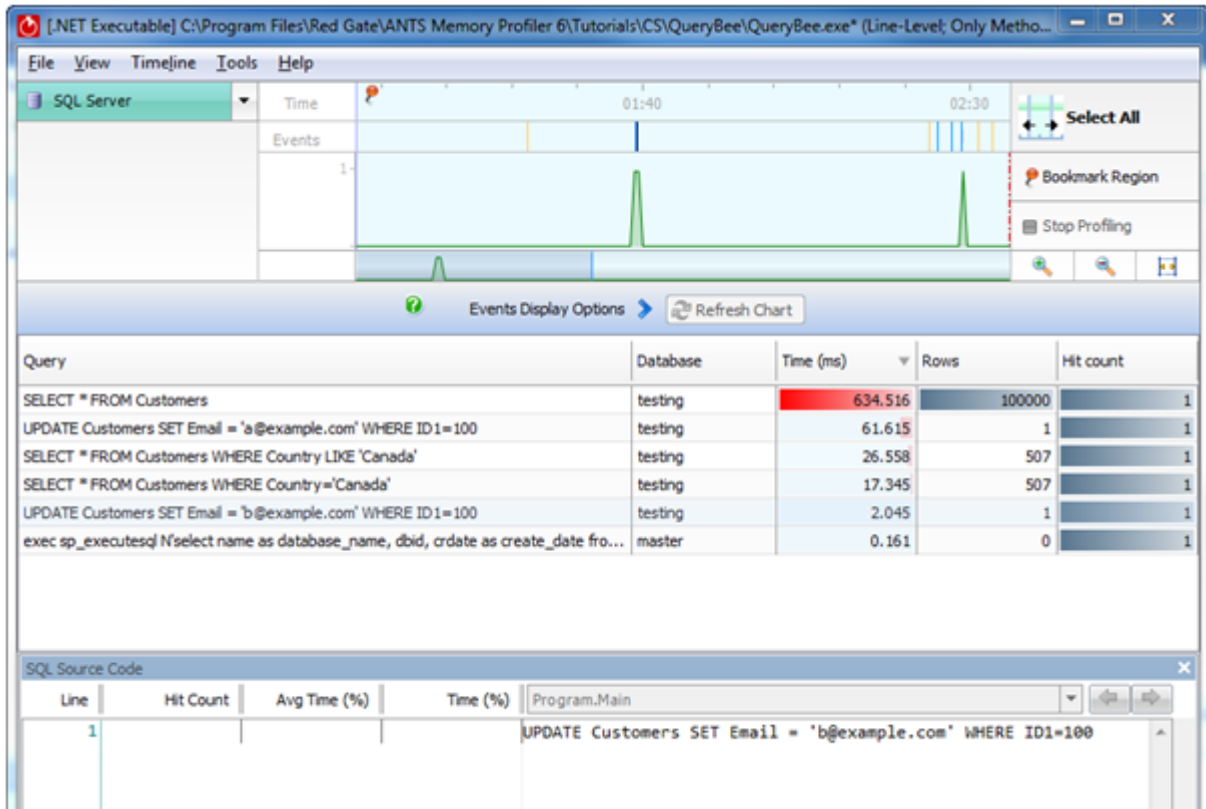
- To show that a primary key will make this query quicker, a primary key is applied to the ID1 field in SQL Server Management Studio:



- Run the following query in QueryBee:  
`UPDATE Customers SET Email='b@example.com' WHERE ID1=100`
- In the ANTS Performance Profiler window, click **Stop Profiling**
- On the **Performance Analysis** menu, set ANTS Performance Profiler to **SQL Server** profiling mode:



ANTS Performance Profiler displays the time taken for each query and the number of rows affected or returned. The LIKE operator is slower than the = operator for text comparison, and that the update query was much quicker with a primary key than without a key.



The examples in this technical paper have been kept deliberately simple. A more complex use case for SQL profiling that you may consider is when complex code generates an SQL query at runtime. In such cases, it might be difficult to check exactly what query has been run, but profiling an application with ANTS Performance Profiler allows you to see immediately how your application has interacted with the SQL server.

Note: SQL Profiling is only available to users of Windows Vista or later, who are profiling a locally-hosted SQL server. It is not possible to use SQL profiling with Microsoft SQL Server Express editions.