IMPACT ANALYSIS

- Refactor large base of C# / SQL Server code and eliminate dead code
- Find database and methods invoking them from the UI handler (button click, checkbox or dropdown)
- Find event handler (see above) from methods invoking a database object (stored procedure or user-defined function)
- Find methods invoking a database object from the DB object name
- All across thousands of lines of code, base classes, even bridging gaps due to delegates or XML files
- Find the impact of modifying a database object or schema
- Find the impact and dependencies of modifying a superclass
- Find these database objects impacted by a UI change

Business cases

- After some time, a code base (Java, C#, C++, Cobol) becomes cluttered with pages, user controls, stored procedures that are no longer used. You want to eliminate them and keep a lean code base.
- Or, you want to identify the UI code that invokes stored procedures directly, bypassing a business rules layer or presenters, or controllers,
- Or, you want to find static and dynamic invocation of methods within external XML files implementing presenters,
- Or, more often, you'll want to know the IMPACT of modifying some stored procedure and identify which event handler in which page triggered its invocation. This, across myriads of business classes, delegates
- Conversely, when modifying UI code, you want to know which stored procedure (s) is/are invoked by some event handler in the UI (page or user control).
- Beware, the fact some stored procedure or user-defined function (udf) is not invoked directly does not necessarily entail it is no longer used! It may well be invoked by other stored procedures and then, should not be eliminated from the code base. Therefore, it is imperative to get the backward (and forward) chain of invocation of stored procedures; otherwise you cannot navigate your code.

- A tool that does all the above, although vital to any company developing code, did not exist. This is why Scientika® designed it.

- Look around, no other product offers such a plethora of useful functions.

Please send us an email so we can send you a white-paper about CodeTracer and SQLTracer:

http://mailto: **info@scientika.com**

Addendum I: **Interoperability**. You know too well that business objects with slightly different names clutter your databases with similar objects, most of the time, not in sync, and often get the wrong data from the wrong table - and, no, two minus don't make a plus in this case! This is why we also refactor databases.

Addendum II: **Localization.** We have designed a method that offers localization-on-the-fly, faster and consuming much less server resources than the bloated Microsoft **.resx files** which are bulk-loaded in memory.

Contact us for our one-day seminar.