

Feature of the Month: Unit and Performance Testing

Visit us and learn more about this and other features at:

<http://www.codexframework.com>

Q: What type of failures does SQL-Hero's Unit and Performance testing catch?

A: Errors that result from actually running an object such as a stored procedure or selecting data from a view are captured. You can provide explicit values for testing or use the Null Run Parameter user-defined property (used for code generation). You can optionally specify expected results which result in failure if these are not returned during testing. You can also set performance targets on test calls: it's a failure if this performance target is not met.

Q: Is testing going to change data in my database?

A: No. Tests are wrapped in transactions that are rolled back.

Q: Can tests be different between different regions such as development and staging?

A: Yes. The user-defined properties that support testing are database specific.

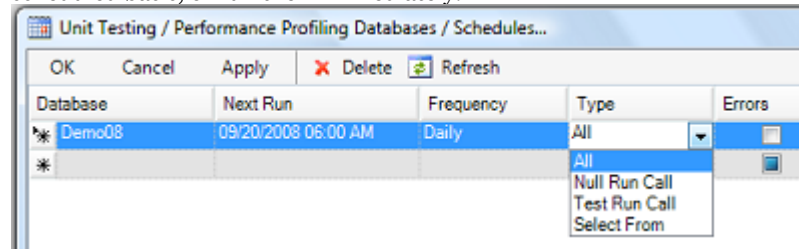
Q: How are tests actually run?

A: They can be run on a schedule or immediately. When done in this way, they're actually running on your SQL-Hero application server, so your machine is left responsive. You can also run execution tests (not performance tests) using MSBuild, such as a step in your code generation.

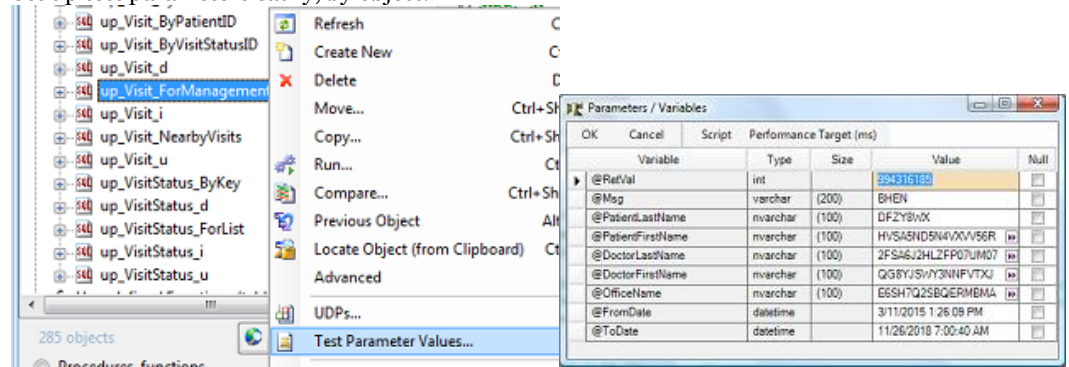
Contact:
admin@codexframework.com

Imagine this situation: you've created a report stored procedure that runs quite quickly – when certain indexes are in place. Someone decides to change these indexes without telling you, and these changes eventually make their way into production. You don't notice this until it's too late: users are complaining about the report taking an hour to run. Having good configuration management processes can help avoid these issues, but the more avenues to catch these types of scenarios – the less time we need to waste tracking down issues.

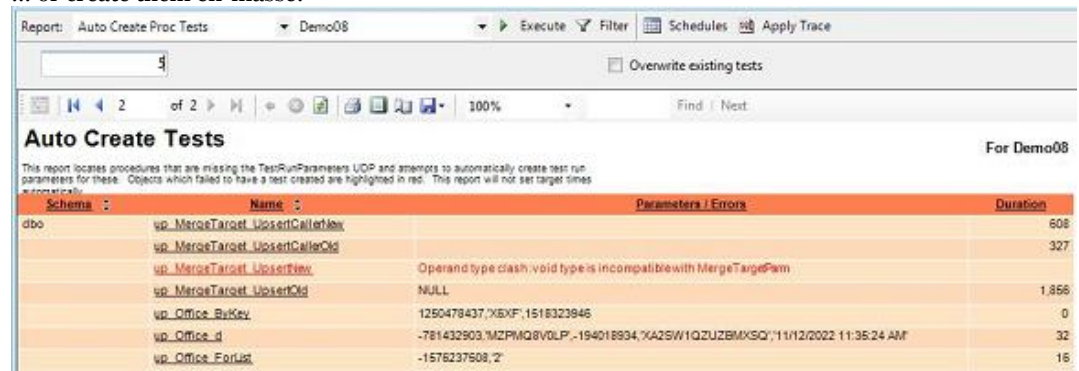
SQL-Hero offers a powerful unit and performance testing tool-set. You can exercise tests on a scheduled basis, or run them immediately:



Set up test parameters easily, by object:



... or create them en-masse:



And now, you can catch failures, reported to you within Visual Studio (Developer / Advanced Edition) or the standalone SQL-Hero executable (Lite Edition):

