Configuring End-to-end Test Scenarios - Overview

SOAtest provides full support for the testing of both Web interfaces and services over multiple protocols. This establishes an integrated framework for "end-to-end" testing; a single test suite can verify operations that cross the messaging layer, the Web interface, the database, and EJBs. Moreover, "Unit tests" can be created as soon as a single "unit of work" is completed, then this test suite can be incrementally enhanced to verify additional components as they are added, test the integration of components within an SOA, and audit end-to-end business processes that span across composite applications.

SOAtest provides a flexible test suite infrastructure that lets you add, organize, and run specialized test scenarios. Each test in a test suite contains a main test tool and any number or combination of outputs (other tools, or special output options). You can run individual tests, or the complete test suite. In addition, you can attach regression controls at the test or test suite level so that you are immediately alerted to unexpected changes. You can move, copy, and delete existing tests and test suites or create new tests manually.

After individual functional tests have been created, they can be leveraged into scenario-based tests without any additional work. Scenario tests allow you to emulate business logic or transactions that may occur during normal usage of the application or service. This also allows you to find bugs that may surface only after a certain sequence of events.

Recommended Workflow

The most common workflow for developing test end-to-end test scenarios is as follows.

As each "unit of work" (service or other logic component) is completed:

1. Use a wizard to automatically generate a project, test suite, and initial test cases.
2. Add additional outputs and tests as needed to cover the functionality that you want to test.
3. Execute the test and verify that it is producing the expected result.
4. If there are problems with the application, diagnose and correct them, then rerun the tests.
5. If the test is not working as expected, adjust test and tool settings as needed, then rerun the tests.
6. Add regression controls or validations once the tested functionality is behaving as expected.

As your test suites grow, combine tests into test scenarios; for example, you can:

- Copy and paste tests from different test suites into a logic order.
- Configure execution options such as test sequence, test relationship, and test flow logic.
- Parameterize tests to use values from data sources or values extracted from
- Use environments to configure a predictable and accessible test bed.