What best practices should I consider when creating ATF tests?

Questions addressed

Page 1:

- What is Automated Test Framework (ATF)?
- What best practices should I consider when creating ATF tests?

Page 2:

- What best practices should I consider when creating ATF tests? (Continued)

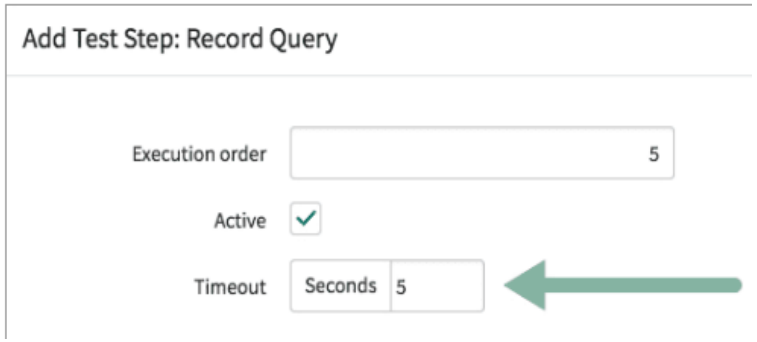
If you have any questions on this topic or you would like to be a contributor to future ServiceNow best practice content, please contact us.

Automated Test Framework (ATF) is a ServiceNow application built for automated testing within a ServiceNow instance to confirm that the instance still works as designed after being upgraded or modified. ATF is active by default in the platform and is available at no extra license cost.

ATF reduces testing time and costs by creating reusable and automated tests that are repeatable & UI independent – reducing overall project costs and schedule.

What best practices should I consider when creating ATF tests?

- **Align ATF to key functional test scenarios –** ATF is intended for functional testing of your specific business processes. Testing should focus on key user flows or process flows within your ServiceNow applications.
- **Impersonate first –** Typically, the first test step of every test impersonates the appropriate user to perform the work. To impersonate a user, use the Create a User test step to create a user with specified roles and groups for the test. The user record gets rolled back after the test completes.
- **Group tests –** Building and running automated test suites allows you to group tests in a specific order to test an application or a group of related features. This way, you're able to run tests and see results as one job.
- **Run tests in parallel –** Reduce testing time by running multiple tests and test suites in parallel. Run tests that create their own data to prevent resource conflicts and data dependencies. You can prevent conflicting tests from running in parallel by marking tests as mutually exclusive. This ensures that tests that require the same data don't run simultaneously.
- **Take a timeout –** User interface (UI) test steps have an intelligent built-in wait mechanism that requires the UI change to complete before the next UI test step proceeds. However, asynchronous updates on the server, such as event processing, workflow updates, and email notifications, might require the test designer to incorporate an additional wait for processing to complete. Setting the timeout on server test steps can also be useful when debugging tests. For example, you could incorporate a 30- to 60-second timeout to manually inspect the state of records before the rollback.



What best practices should I consider when creating ATF tests? (Cont.)

What best practices should I consider when creating ATF tests? (Continued)

- **Keep it short –** Test a discrete set of items during testing. This makes it easier to build tests and evaluate any failures. It is easy to string multiple tests together as part of a suite and you can even use test suites hierarchically. This allows you to perform additional testing should an early step in a test fail.
- **Avoid test step repetition –** Don't repeat testing for the same UI functionality in multiple tests. If other tests require similar functionality, simulate it using server test steps rather than repeating the same UI test steps. Server test steps perform faster.
- **Test the outcome –** Focus on the outcome to verify rather than testing each point along the way. Keep this in mind when verifying business rules or Flow Designer processes. When testing flows, build the conditions to trigger the flow in the test, then check the flow outcome for the expected result.
- **Consider access control rule (ACL) security –** When debugging unexpected test errors, a common mistake relates to security authorization. Based on the roles of the impersonated user, a test step might fail. By default, ACL security is enforced for record inserts, updates, and deletes. Sometimes this setting must be disabled for test steps performing actions to set up a test. Other test steps actually performing the verification might require it to be enabled.
- **Validate record updates –** Include a Record Validation test step after every Record Update test step to ensure the record was actually updated. A Record Update appears to always pass even if the record was not updated for any reason, such as being rejected by a data policy. This can lead to confusing failures in subsequent test steps. It can also mislead you in trying to troubleshoot the wrong test step.
- **Test functionality instead of data –** Rather than test every possible combination of a data-driven feature, create one or two tests focusing on the functionality. Then verify that the values change as intended without spending too many cycles on all the possible combinations of a vast data matrix.
- **Approach conditional testing differently –** When you want to perform ATF test steps based on given criteria or branch a test along different paths, create a separate test for each condition.

Related resources

- Success Quick Answer – When and how should I use ATF?
- Product Docs -- Automated Test Framework overview
- Now Community – Best practices for using ATF
- Getting started with ATF
- ATF Fundamentals eLearning series
- New for developers in Paris: Overview of Automated Test Framework changes
- Success Playbook – Perform ServiceNow upgrades quicker and more effectively
- Success Checklist – Plan for upgrades at least once a year (PPT)
- Upgrading to a new release