

When and how should I use Automated Test Framework (ATF)?

Questions addressed

Page 1:

- What is Automated Test Framework (ATF)?
- Where can I find ATF?
- Why use ATF to test your instance?

Page 2:

- When and how should I use ATF?

Page 3:

- What is the best practice process for using ATF?

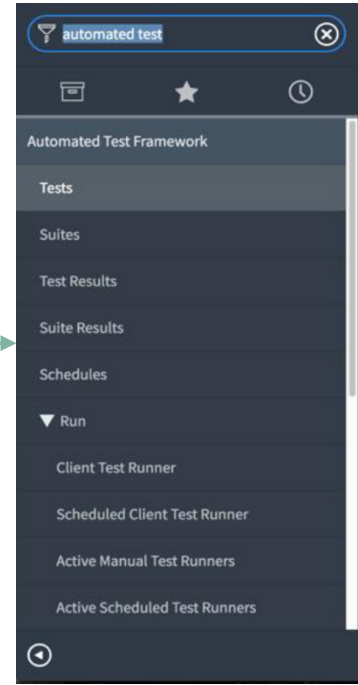
[Automated Test Framework \(ATF\)](#) is a ServiceNow® application used to automate the tests performed to verify applications, customizations, or configurations. ATF gives you the tools to [create and run automated tests](#) on your ServiceNow non-production instance.

ATF provides functional testing that allows actions such as creating records, setting field values, and checking results of field values. If a test fails, ATF indicates that the result doesn't meet the expected results of the test and includes details of what the underlying problem may be. Also, **ATF is free** with the Now Platform® and uses the same development tools your developers are familiar with.

Search for "Automated Test Framework" in the application menu to find it. →

Why use ATF to test your instance? ATF reduces upgrade and development time by replacing manual testing with automated testing. Compared to other test methods, ATF:

- **Avoids time-consuming and repetitive testing** – Just set up the tests and reuse them every time you upgrade or modify your instance.
- **Avoids user error** – Since the tests are automated, they're performed exactly the same way every time you run them.
- **Allows you to create reusable tests** – Once you create ATF tests, you can reuse them for future upgrades and other releases.
- **Speeds up test creation** – Use either a [quick start test](#), which is a ServiceNow-provided test that you can copy and modify to meet your unique business process needs, or create [test templates](#) to add predefined lists of steps to existing test plans.
- **Increases speed to market** – You'll upgrade faster and get your value-added features and enhancements to your users faster.
- **Avoids breaking tests related to user interface (UI) changes** – ATF is independent of UI changes, so tests do not break when the UI is changed, which is a common issue with third-party automated testing tools.
- **Promotes confidence in your platform stability** – It delivers a quality product with ServiceNow.



When and how should I use Automated Test Framework? (Continued)

When and how should I use ATF?

You should run ATF tests to confirm that the instance still works as designed whenever you upgrade, modify an instance (e.g., customize or make field changes), or have internal development releases. This includes running tests on any scoped apps, custom apps, or out-of-the-box products.

With ATF, you can run an entire suite of tests and, in moments, analyze the results to determine if remediation is required. You can use the same tests again for your next release or upgrade if you keep the tests in sync with the configurations and customizations they test.

ATF is intended to test the changes you make to your instance, but not to test base ServiceNow functionality.

Use this chart to better understand what type of testing is recommended for ATF.

Recommended	Not recommended
ATF is intended for functional testing of business logic , so it tests the specific business processes that you manage in your ServiceNow instance. For example, every organization has a specific change management process. ATF allows you to build a test where you can automatically create a change request and push it through its lifecycle to ensure that your process still works.	You should never use ATF tests or any other functional testing tools in your production instance.* For example, a test can impersonate a user with extensive security access or change data that may be designed to trigger actionable events, like sending an email to your entire customer base.
ATF targets browser compatibility testing across multiple browsers and operating systems. Write an ATF test once, and then verify the same functionality works as expected for all supported environments.	ATF is not intended for testing every single UI component of your ServiceNow instance. For example, you don't need to test the magnifying glass icon on a reference field. Think about what's unique in your instance and what's common across every instance.
ATF is effective for regression testing. Create comprehensive testing plans to ensure existing processes still work as expected following an upgrade or development cycle. Run the same comprehensive testing plan before and after major changes to get a baseline. ATF provides assurance that the same tests will run in the same manner, allowing routine instance checks to be performed whenever necessary.	ATF can be used but is not recommended for unit testing. For example, since you'll need to update ATF tests every time a feature changes, don't build ATF tests for features that are still changing frequently. It's better to create tests once feature changes have slowed down to prevent extra work.
ATF can be integrated with Jasmine , an open source testing framework for JavaScript. Jasmine tests are called to verify server-side functionality from the steps of an ATF test.	ATF is not a load testing or performance testing application.

***ATF tests should only be run in sub-production.** If you clone your sub-production instance from production, consider moving ATF tests to your production instance using [update sets](#) so you don't lose the tests from your sub-production instance before they're cloned from production.

When and how should I use Automated Test Framework? (Continued)

What is the best practice process for using ATF?

1

Identify essential functionality per process/application.

Focus on the functionality that's specific to your ServiceNow instance. When determining what's essential, consider the exposure to risk. Weigh the detail of the validation against the ongoing [test](#) maintenance required.

2

Create tests to validate essential functions.

Begin creating tests once any additional changes to the new feature you want to test will be minimal. If you develop new functionality and ATF tests in tandem, realize any additional functional changes could result in additional ATF development effort.

3

Package tests into suites.

Organize your tests into [test suites](#) that can be run together. Use an existing test suite or create a new one. The same test can be grouped into multiple test suites.

4

Run or schedule test execution.

ATF test suites can either be run on demand or scheduled. ATF tests are only run on demand. The [test steps](#) within a test are run on the client or the server. Test steps performed on the client UI are executed by a [client test runner](#) within a separate browser window. Test steps performed on the server execute against the instance.

5

Monitor and fix issues.

Once tests are run, review the [test results](#). Detailed test results are provided at the test suite, test, and test step level. Runtime screenshots are included in test results to help monitor and debug any issues.

6

Maintain ATF tests.

Whenever changes (e.g., customizations or field changes), are introduced to an instance, evaluate and update ATF tests to reflect the latest changes.

Related resources

- [Success Quick Answer – What best practices should I consider when creating ATF tests?](#)
- [Product documentation – Automated Test Framework Overview](#)
- [New for Developers in Madrid: 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](#)
- [ATF Fundamentals eLearning series](#)
- [Best practices for using ATF](#)
- [Getting started with ATF](#)

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 at best.practices@servicenow.com.