Oracle® Communications Solution Test Automation Platform

Release Notes

Release 1.26.1.0.0
G48768-01
April 2026

Oracle Communications Solution Test Automation Platform Release Notes, Release 1.26.1.0.0

G48768-01

Contents

About This Content

This document describes how to implement and use Oracle Communications Solution Testing Automation Platform.

Audience

This document is intended for anyone who installs, configures, administers, or customizes Solution Testing Automation Platform.

1

Release Notes

This document provides the release notes for Oracle Solution Test Automation Platform (STAP) 1.26.1.0.0.

This document contains the following sections:

New Features

This section describes the new features in STAP release 1.26.1.0.0.

Intelligent UI Search

Intelligent Search in the UI (STAP Dashboard) provides test recommendations to help you create new test jobs rapidly by re-using existing test artifacts. You use keyword-based search to find available test artifacts, and the search results help you create new test jobs quickly. For more information, see "Intelligent Search"

Logical Operators

You can now use logical operators in BDD scenarios to define and evaluate multiple conditions in validation and save steps. **%EVAL_CONDITION()** evaluates expressions with AND, OR, and NOT, supports operator precedence, and allows parentheses for explicit grouping. **%IF_ELSE()** saves one of two values based on a specified condition during the save operation, enabling condition-based value selection in workflows such as subscription management. For more information, see "Logical Operators".

Using Reference Step

Define reusable steps in **.stepref** files under **referenceSteps.home** and assign each a unique **ReferenceStepId**. In a **.case** file, reference the predefined step by **ReferenceStepId** using **When, Then, or And**. At runtime, STAP expands the referenced step at the usage point and merges any overridden or additional **Data, Validate, and Save** entries; the keyword comes from the usage step. For more information, see "Using Reference Step".

Using ForEach in a Test Case

Run a single Case multiple times with **ForEach: $filename.csv**, where each CSV row drives one execution of the Case. Reference CSV values with **${&<parameter>}** to map header names to runtime values. This enables data-driven execution without duplicating test logic. For more information, see "Using ForEach in a Test Case".

Common Data Files

Centralize reusable data files by setting **referenceFiles.home=${WORKSPACE}/files/** and reference them with **$FILEREF(<filename>)**. STAP resolves **$FILEREF()** paths relative to **referenceFiles.home** and replaces the reference with the file content at runtime. This supports shared payloads, scripts, and configuration files across scenarios. For more information, see "Common Data Files".

Supported UI Actions in UI Automation Plug-in

STAP UI Automation Plug-in adds supported UI actions as modular, scriptable commands for UI automation.

It provides **Browser Actions, Form Actions, Save Actions, and Validation Actions** to express common operations navigation with **$open**, browser controls (**$back, $forward, $refresh, $alert, $waitFor**), form interactions (**$input, $click, $doubleClick, $selectFromDropdown**, scrolling, key presses, **$uploadFile**), context and UI value capture with **Save** and **$VALUE**, and validations with **$url, $title, and $visible**. This capability converts Selenium WebDriver logic into concise, reusable, data-driven steps for end-to-end UI workflows. For more information, see "Supported UI Actions".

Stop Support for Running Jobs in Dashboard

STAP supports stopping a running job from the Dashboard screen. Use the stop icon in the Running Jobs section to initiate a safe and controlled termination, with a confirmation dialog and graceful shutdown of all processes. For more information, see "Stopping a Running Job".

Eval Function Support in Validate Section

STAP enables the **Eval** function in the **Validate** section, allowing users to use expression evaluation directly during validation within STAP workflows. For more information, see "Numeric Function: Evaluate to Process Arithmetic Expressions".

ARRAY_UNIQUE in Array Utility Functions

STAP adds **ARRAY_UNIQUE** in Array Utility Functions to derive a new array that contains only unique values from an input array. This supports validation against a distinct set of elements when response data includes duplicate entries. For more information, see "Array Operators"

Update Support for Scenario Publish Using the Command-Line Interface

STAP supports updating an existing scenario during **--publish -scenario** when **update=YES** is set in the scenario **publish.properties** file. On update, STAP marks the existing scenario as inactive, generates a new scenario ID, and automatically updates all existing jobs to use the new scenario ID; after a successful update, STAP resets update to **NO**. For more information, see "Publishing Actions, Scenarios and Environments Using the Command-Line Interface".

Kafka Connection Authentication Support (Basic and OAuth2)

Kafka Connection supports two types of authentications: Basic and OAuth2. Configure authentication in the Kafka environment properties using **authorization=YES** and **authorization.type=basic** or a**uthorization.type=oauth2**. For more information, see "Kafka".

REST Plug-in Retry Count Support

REST Plug-in supports **retryCount** in the environment configuration to control the number of automatic retries for failed REST API calls when the failure returns a server-side error (HTTP 5xx). For more information, see "REST Plug-in".

Environment Variables for Variable Injection

STAP supports Environment Variables (Externalized Configuration Inputs) defined in the environment file as **variable.<name>=<value>** and injected at runtime into scenarios. Access these values in steps using **${ENV_<name>}** to vary environment-specific data without changing scenario files. For more information, see "Using Variables".

Known Problems

This section lists the known problems in this release of STAP.

- **Duplicate Environment Names are Allowed**
  - The system allows multiple environments to be created with the same name.
  - **Workaround:** Before creating a new environment, search for existing ones to avoid duplicates.
- **Only Non-Referenced and Unused Jobs can Be Deleted**
  - Jobs with execution history or reports cannot be deleted in this release to retain historical data of jobs.